NASnews

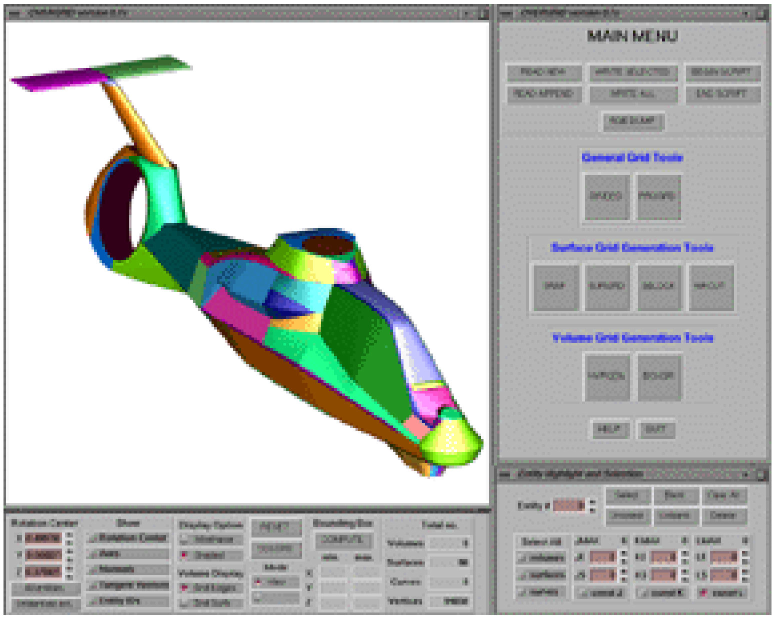
May–June 1998 • Volume 3, Number 3

In this issue...

Innovative Software Tool Streamlines Overset Grid Generation

For many years, limitations in technology have exasperated engineers who face the complex and time-consuming task of producing computational grids needed for flow simulation problems in design and development. Some of those engineers are now praising a new software tool called OVERGRID, which streamlines the grid generation process considerably.

To The Article...



The OVERGRID user interface has four main windows. The Graphics window (upper left) displays geometry and grids. The Main Menu (upper right) performs I/O and gives users access to grid generation modules. The Controls window (lower left) contains widgets for modifying attributes of the display. The Entity Selection window (lower right) allows picking, blanking and deleting entities. *Graphic by William Chan*.

NASnews
Main Menu
May–June 1998 • Volume 3, Number 3

Innovative Software Streamlines Overset Grid Generation

by **William M. Chan**

For many years, limitations in technology have exasperated engineers who face the complex and time-consuming task of producing computational grids needed for flow simulation problems in design and development. Some of those engineers are now praising a new software tool called OVERGRID, which streamlines the grid generation process considerably.

OVERGRID provides much faster results -- in about a quarter of the time it used to take to create high-quality overset grids -- and requires much less effort by the user. Applications include commercial and military aircraft, space launch and return vehicles, and marine ships and submarines, among others.

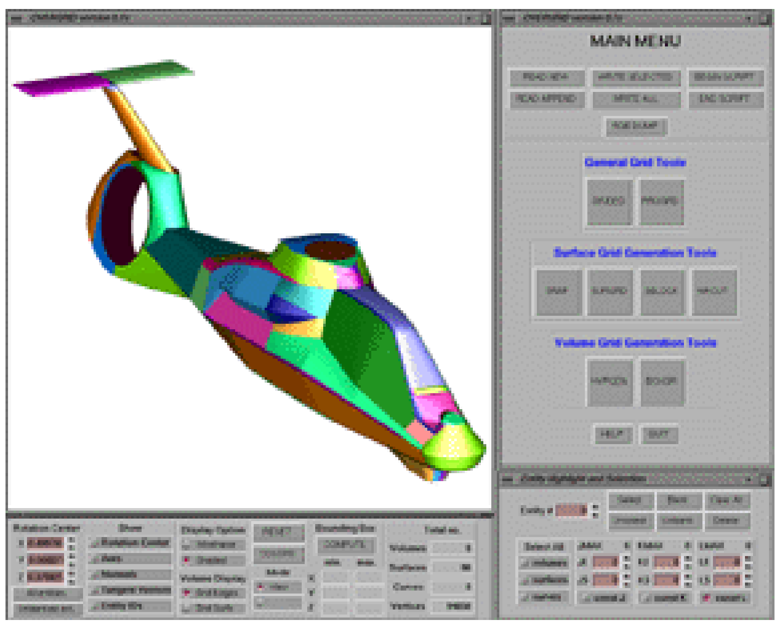
In the past, users had to apply several different, independently developed software utilities to generate overset surface and volume grids for complex configurations. The procedure involved multiple steps and software tools, and the results from each step had to be loaded into a visualization package (such as PLOT3D or FAST) for inspection before advancing to the next step.

With OVERGRID's release last September, users can now do all their grid generation work and visualization under one environment. Through a graphical interface, users can execute each step and inspect results before moving on to the next step, and are able to observe the effects of altering various parameters immediately. Moreover, the different steps and parameters specified by the user are automatically written to a script that can be modified and replayed rapidly in batch mode.

OVERGRID was developed in the Rotorcraft Aeromechanics Branch at NASA Ames Research Center, with funding from the DoD High Performance Computing (HPC) Modernization Program under the Common HPC Software Support Initiative.

Software Overview

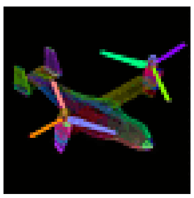
OVERGRID's graphical user interface (GUI) gives users access to various overset grid generation modules, as well as the ability to visualize the geometry and grids at each step of the process. The modules include grid manipulation utilities as well as several surface and volume grid generation tools tailored for overset grids. Input parameters for the modules can be created and modified in OVERGRID. Execution of these modules is controlled from OVERGRID using batch mode calls. The results are automatically loaded into OVERGRID for visualization.



The OVERGRID graphical user interface has four main windows. The Graphics window (upper left) displays the geometry and grids. The Main Menu window (upper right) performs I/O and gives users access to various grid generation modules. The Controls window (lower left) contains widgets for modifying various attributes of the display. The Entity Selection window (lower right) contains widgets for picking, blanking and deleting entities. *Graphic by William Chan*.

OVERGRID is written in C, OpenGL, Tcl/Tk, and Fortran -- making it portable among wide variety of platforms, from workstations to personal computers. OpenGL is used to render the geometry and grids in the Graphics window of the interface. The Tool Command Language and associated Toolkit (Tcl/Tk) provides a quick, simple, and flexible way to build widgets such as buttons, entries, and sliders for the GUI in a compact code. OpenGL rendering is achieved with a Tk widget called Togl, developed by Brian Paul at Avid Technology and Ben Bederson at the University of Maryland, College Park.

Visualization Capabilities



Surface normals, tangent vectors and entity identification numbers
Figure1
Figure2
Figure3
Figure4

With OVERGRID's interface, users can choose to display the geometry and surface grids in wireframe or smooth-shaded mode. Entities in the Graphics window are manipulated using mouse transforms similar to those used in PLOT3D.

In addition, a drag-and-zoom box performs local zooming and rotation-center modifications. Various attributes, including surface tangent vectors, surface normals, and entity identification numbers, can be displayed by toggling widgets in the Controls window.

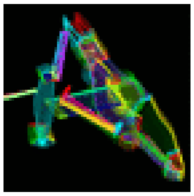
The Entity Selection window contains widgets for selecting and unselecting, blanking and unblanking, and deleting entities. It also houses controls for sweeping through surface slices of volume grids in the three index directions.

Grid Generation Process in a Nutshell

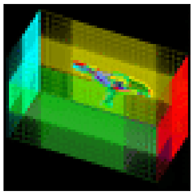
Hyperbolic surface grid generation using SURGRD
Figure1
Figure2
Figure3
Figure4
Figure5

OVERGRID reads input geometry or grid files in PLOT3D format. The surface grid generation steps involve abstract decomposition of the surface domain, identification and extraction of seam curves, redistribution of points on the seam curves, and generation of seam surface grids from the seam curves. Seam curves occur along surface discontinuities, intersection lines between components, and high-curvature contours.

Hyperbolic or algebraic methods are used to generate surface grids for domains bounded by one or multiple seam curves, respectively. Body-conforming hyperbolic volume grids are then "grown" from each of the overlapping surface grids. Finally, this collection of hyperbolic volume grids is embedded in a Cartesian box grid with uniform core spacing and stretched outer layers. See "OVERGRID in Action" for a complete list of the software's grid generation capabilities.



Body-conforming hyperbolic volume grid generation using HYPGEN
Figure1
Figure2
Figure3
Figure4
Figure5



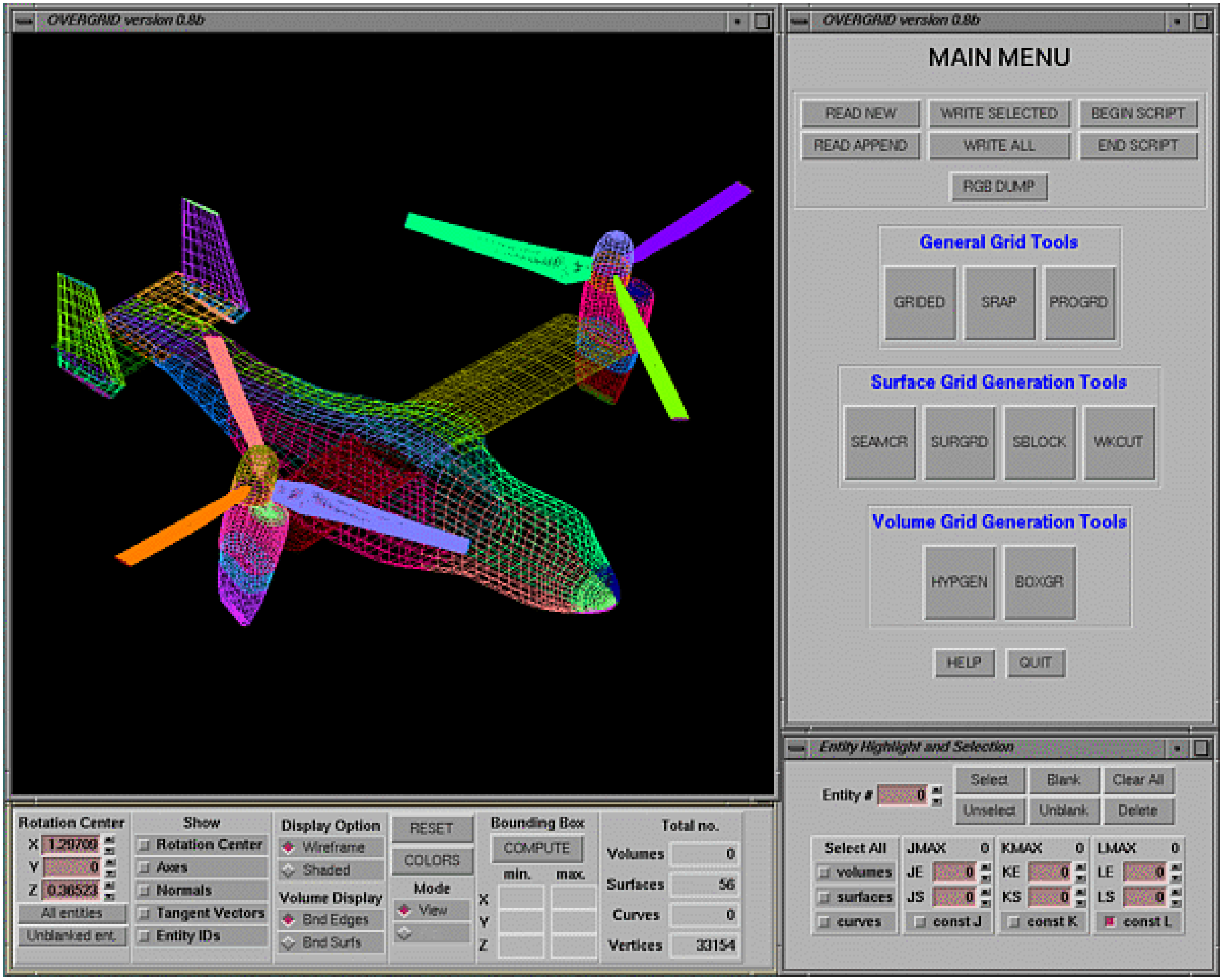
Stretched Cartesian box grid generation using BOXGR
Figure1
Figure2

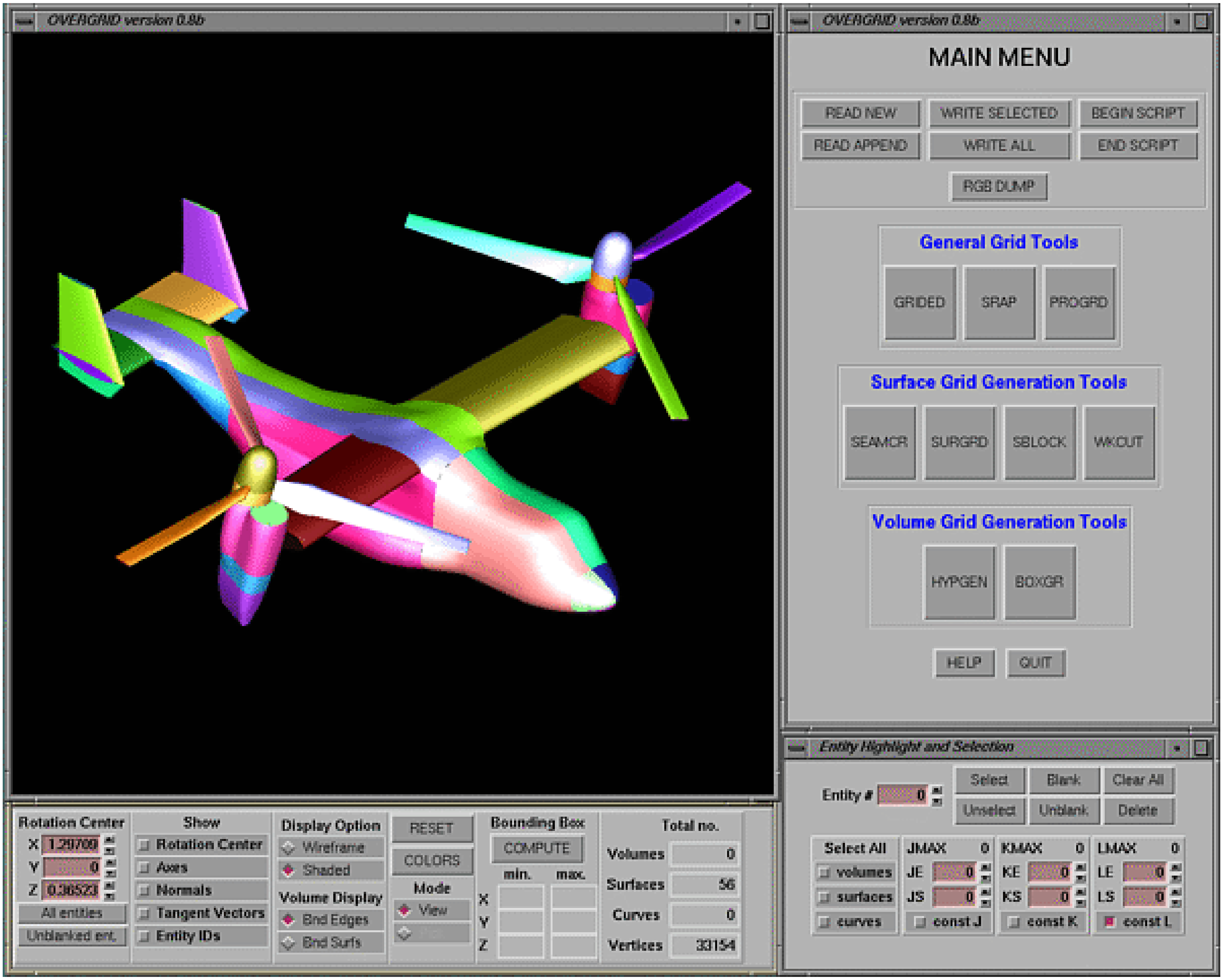
Available to U.S. Organizations

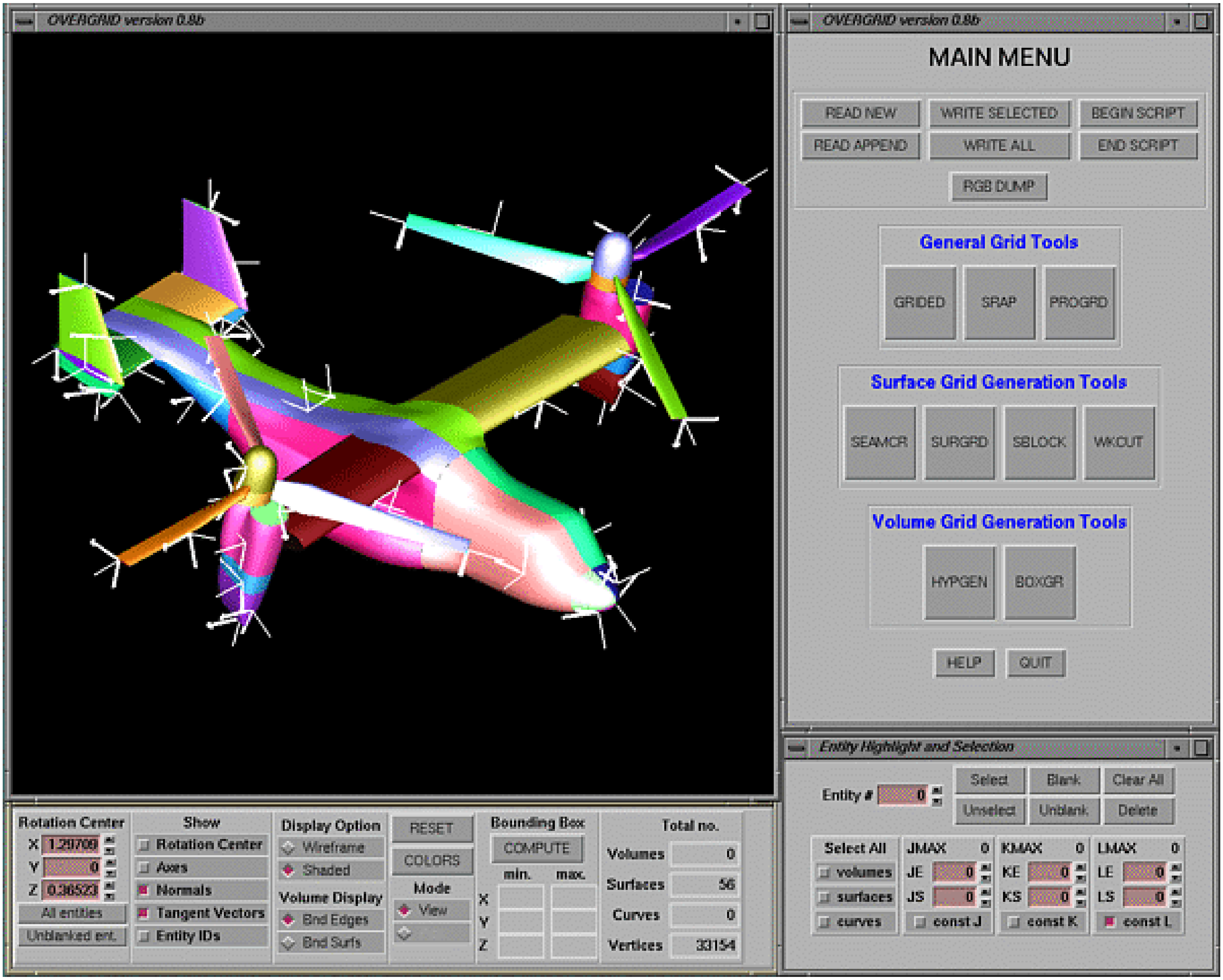
OVERGRID is part of the Chimera Grid Tools software package (also developed at Ames), which includes many grid generation and manipulation modules and some flow solver postprocesssing tools. It is available for internal use only to companies, universities, and government labs in the U.S. To get the code, requesting organizations must first complete Space Act Agreement documents and obtain approval from Ames Research Center management. (*Ames researchers can send email to the author*).

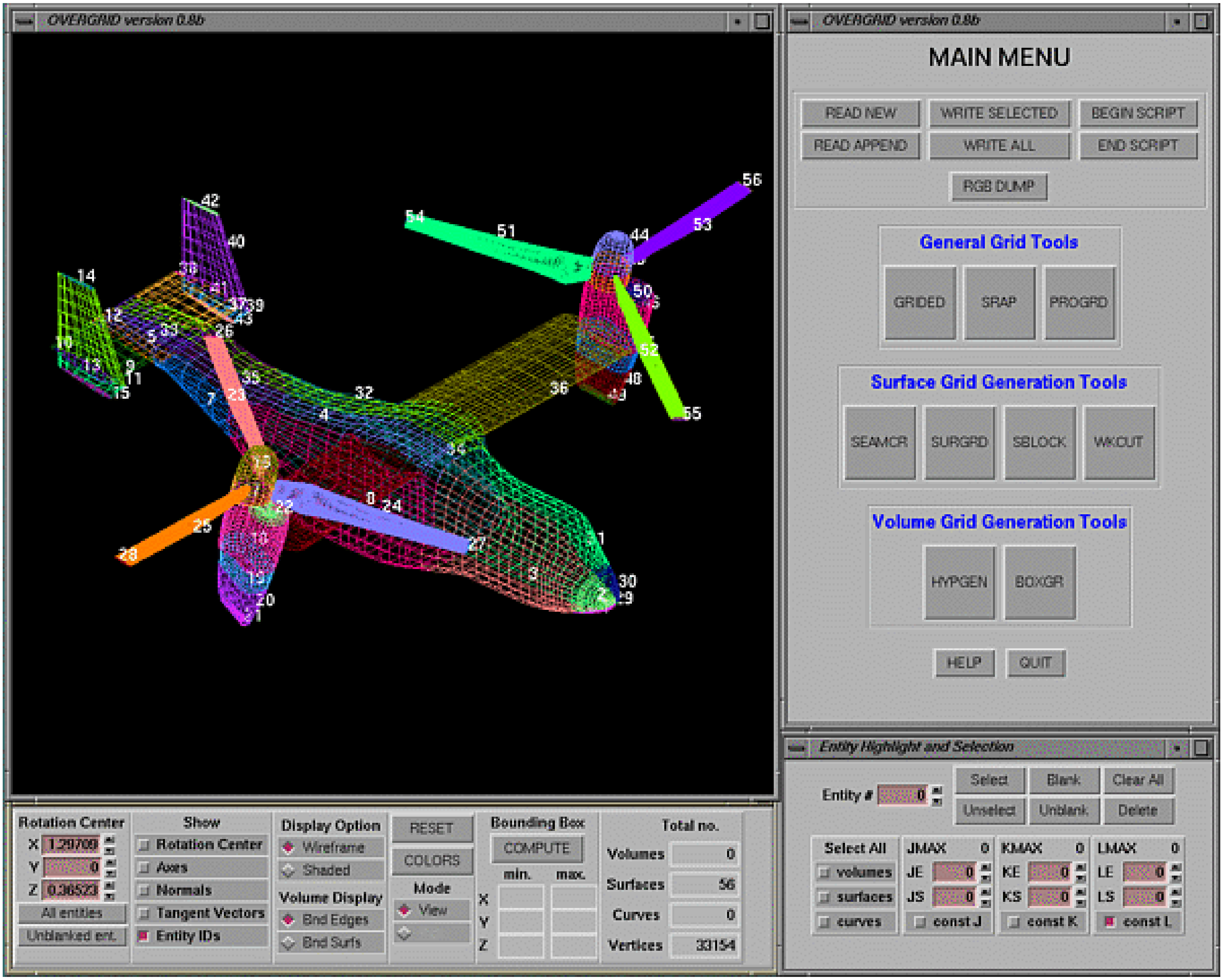
Future Work Includes NURBS

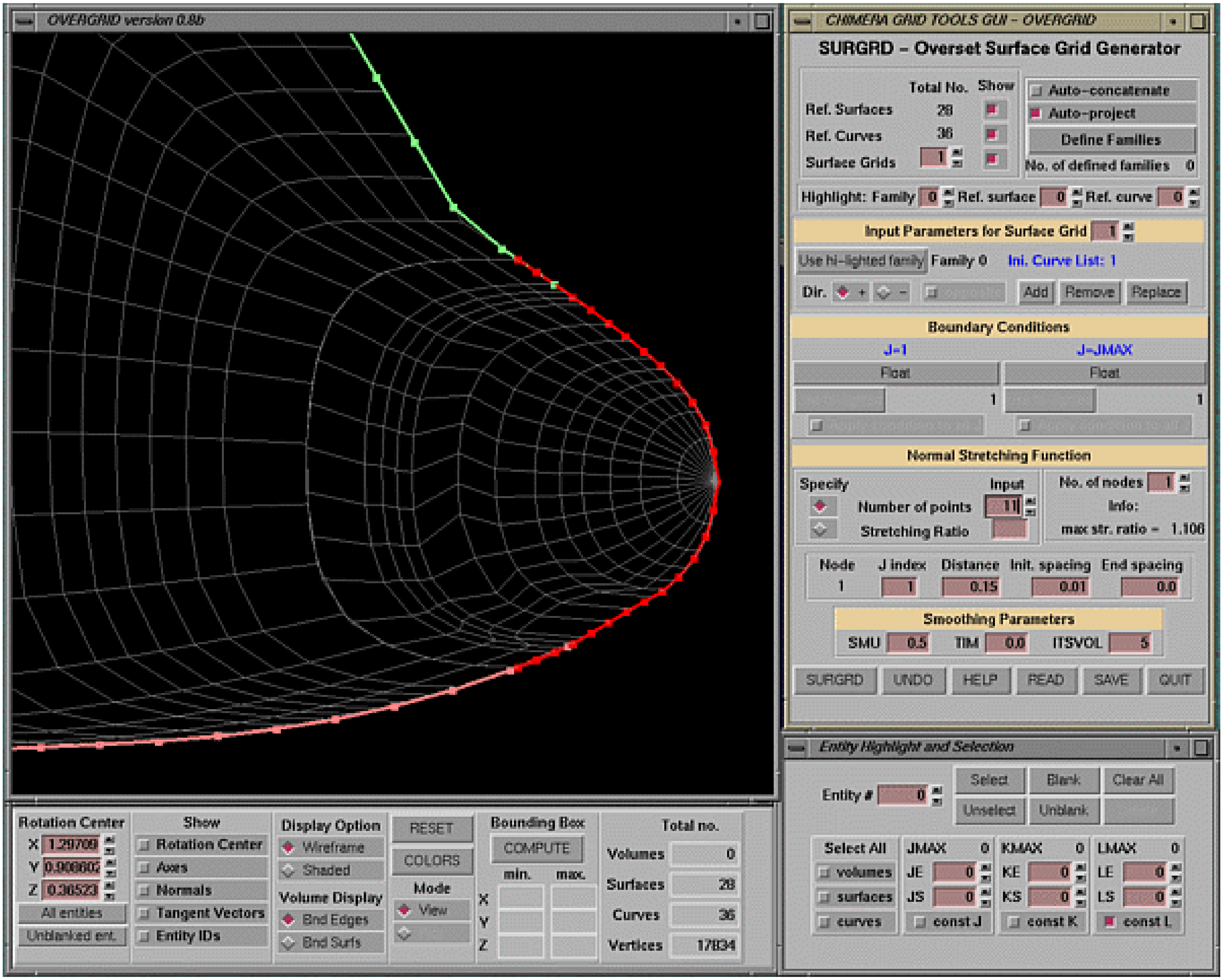
New features and improvements are frequently added to OVERGRID. Work in progress includes automatic seam curve detection and the use of Non-Uniform Rational B-Spline (NURBS) surfaces -- a common format for surface geometry description -- and trimmed NURBS surfaces as input geometry definition. More information on OVERGRID is available on the Web.

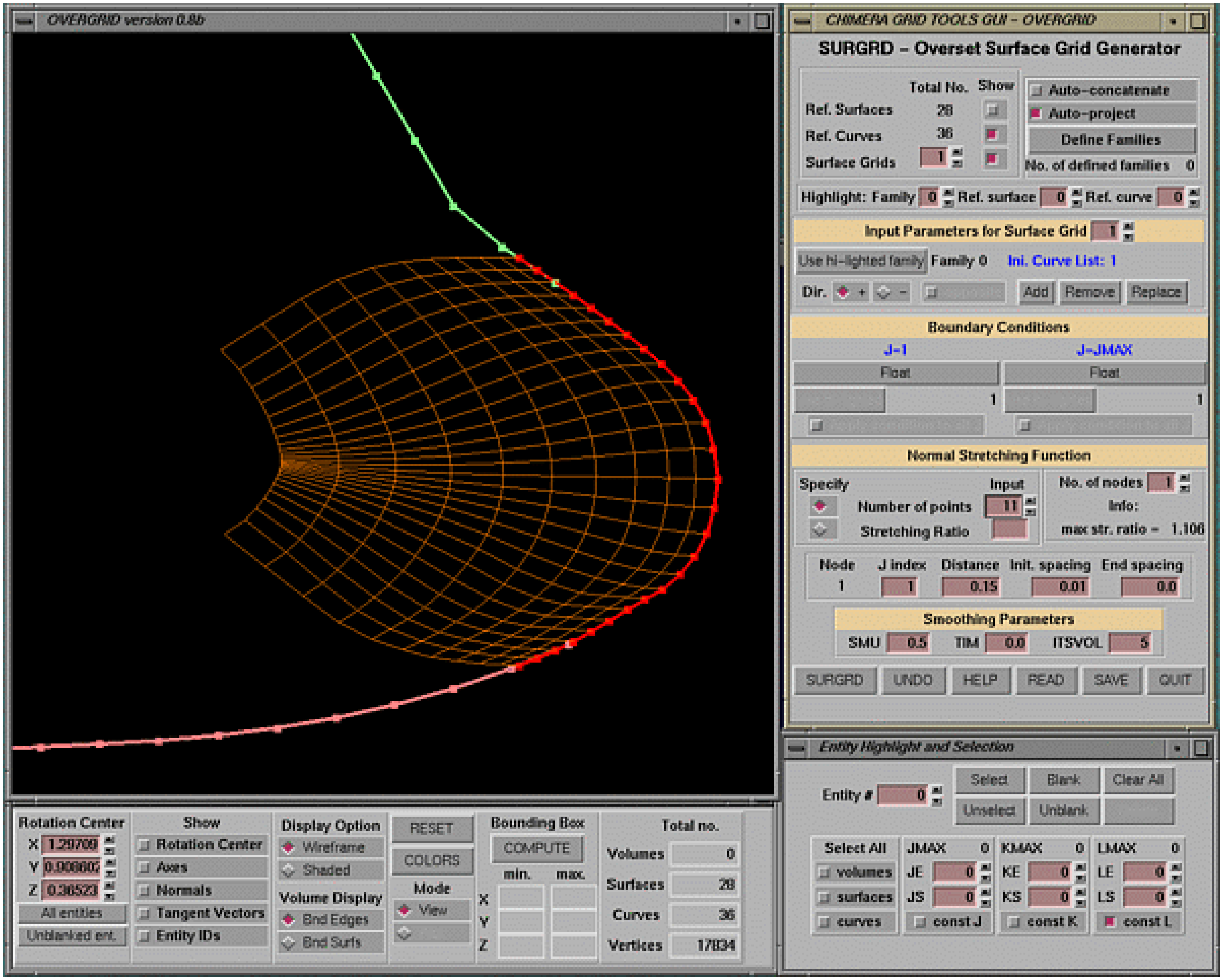

OVERGRID version 0.8b
MAIN MENU
READ NEW
WRITE SELECTED
BEGIN SCRIPT
READ APPEND
WRITE ALL
END SCRIPT
RGB DUMP
General Grid Tools
GRIDED
SRAP
PROGRD
Surface Grid Generation Tools
SEAMCR
SURGRD
SBLOCK
WKCUT
Volume Grid Generation Tools
HYPGEN
BOXGR
HELP
QUIT
Entity Highlight and Selection
Entity #
Select
Blank
Clear All
Unselect
Unblank
Delete
Select All
volumes
surfaces
curves
JMAX 0
JE 0
JS 0
const J
KMAX 0
KE 0
KS 0
const K
LMAX 0
LE 0
LS 0
const L
Rotation Center
X 1.29709
Y 0
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min.
max.
X
Y
Z
Total no.
Volumes 0
Surfaces 56
Curves 0
Vertices 33154

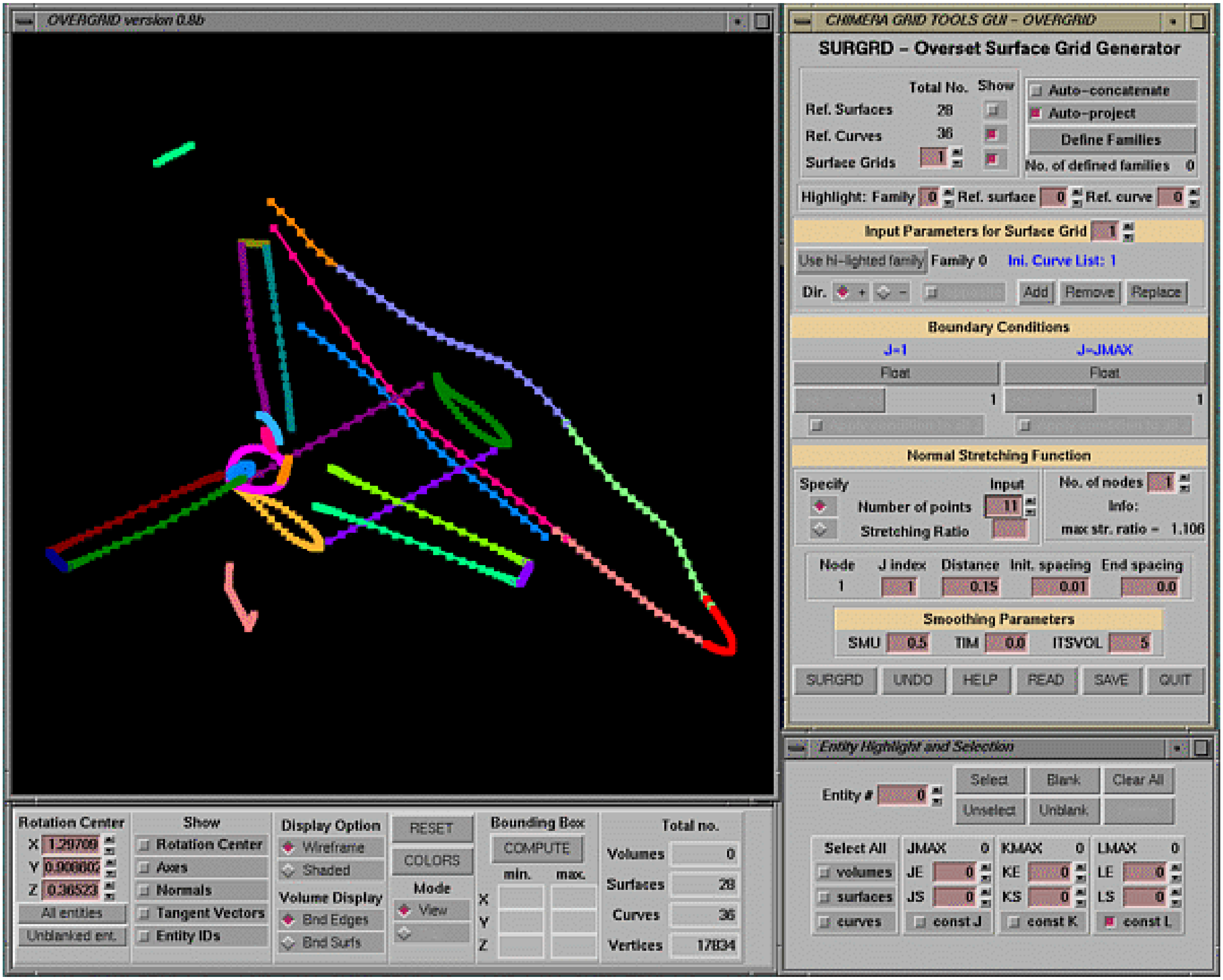

OVERGRID version 0.8b
MAIN MENU
READ NEW
WRITE SELECTED
BEGIN SCRIPT
READ APPEND
WRITE ALL
END SCRIPT
RGB DUMP
General Grid Tools
GRIDED
SRAP
PROGRD
Surface Grid Generation Tools
SEAMCR
SURGRD
SBLOCK
WKCUT
Volume Grid Generation Tools
HYPGEN
BOXGR
HELP
QUIT
Entity Highlight and Selection
Entity #
Select
Blank
Clear All
Unselect
Unblank
Delete
Select All
volumes
surfaces
curves
JMAX
JE
JS
const J
KMAX
KE
KS
const K
LMAX
LE
LS
const L
Rotation Center
X 1.29709
Y 0
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min.
max.
Total no.
Volumes 0
Surfaces 56
Curves 0
Vertices 33154

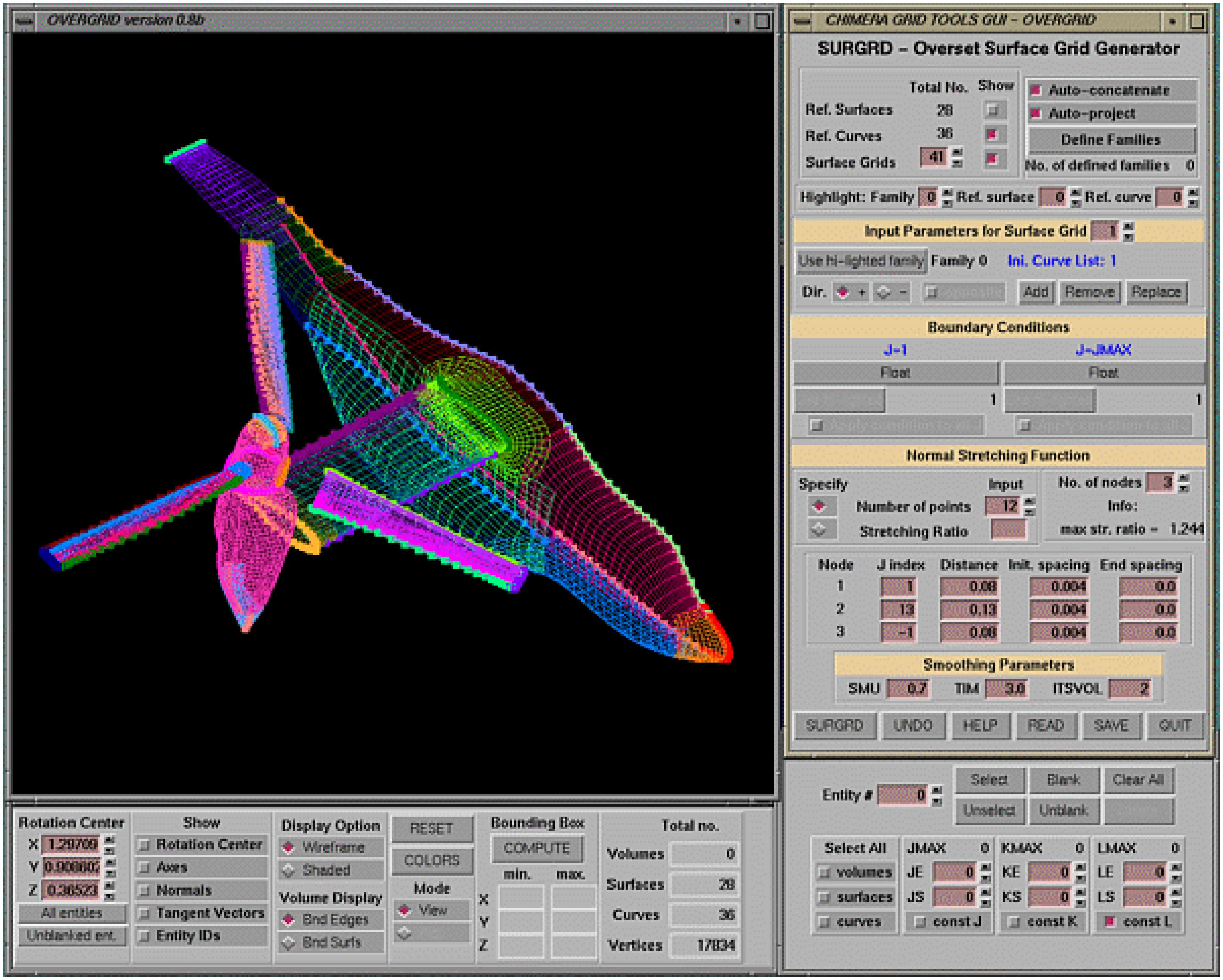

OVERGRID version 0.8b
OVERGRID version 0.8b
MAIN MENU
READ NEW
WRITE SELECTED
BEGIN SCRIPT
READ APPEND
WRITE ALL
END SCRIPT
RGB DUMP
General Grid Tools
GRIDED
SRAP
PROGRD
Surface Grid Generation Tools
SEAMCR
SURGRD
SBLOCK
WKCUT
Volume Grid Generation Tools
HYPGEN
BOXGR
HELP
QUIT
Entity Highlight and Selection
Entity #
Select
Blank
Clear All
Unselect
Unblank
Delete
Select All
volumes
surfaces
curves
JMAX 0
JE 0
JS 0
const J
KMAX 0
KE 0
KS 0
const K
LMAX 0
LE 0
LS 0
const L
Rotation Center
X 1.29709
Y 0
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min.
max.
X
Y
Z
Total no.
Volumes 0
Surfaces 56
Curves 0
Vertices 33154

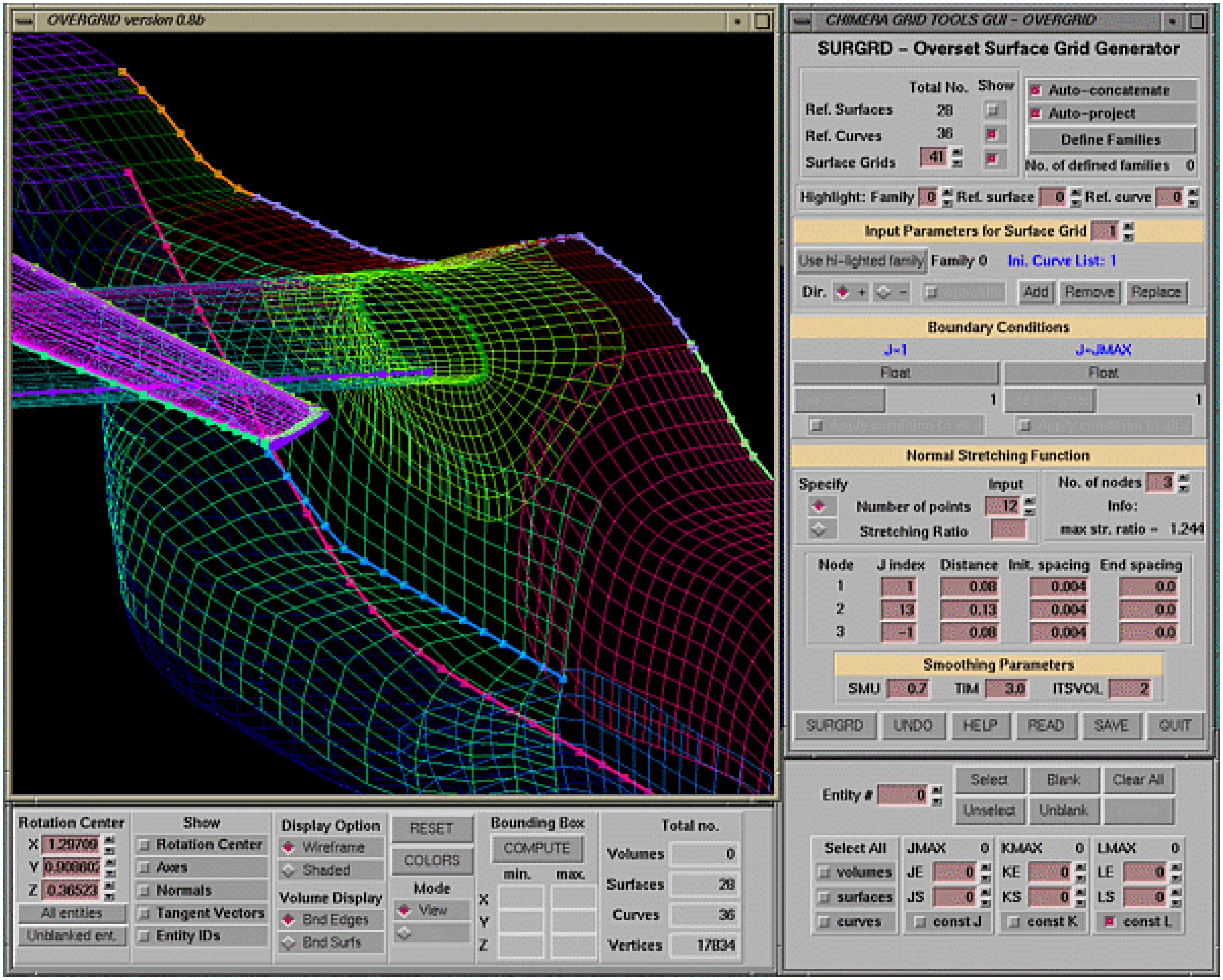

OVERGRID version 0.8b
OVERGRID version 0.8b
MAIN MENU
READ NEW
WRITE SELECTED
BEGIN SCRIPT
READ APPEND
WRITE ALL
END SCRIPT
RGB DUMP
General Grid Tools
GRIDED
SRAP
PROGRD
Surface Grid Generation Tools
SEAMCR
SURGRD
SBLOCK
WKCUT
Volume Grid Generation Tools
HYPGEN
BOXGR
HELP
QUIT
Entity Highlight and Selection
Entity #
Select
Blank
Clear All
Unselect
Unblank
Delete
Select All
volumes
surfaces
curves
JMAX 0
JE 0
JS 0
const J
KMAX 0
KE 0
KS 0
const K
LMAX 0
LE 0
LS 0
const L
Rotation Center
X 1.29709
Y 0
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min.
max.
X
Y
Z
Total no.
Volumes 0
Surfaces 56
Curves 0
Vertices 33154

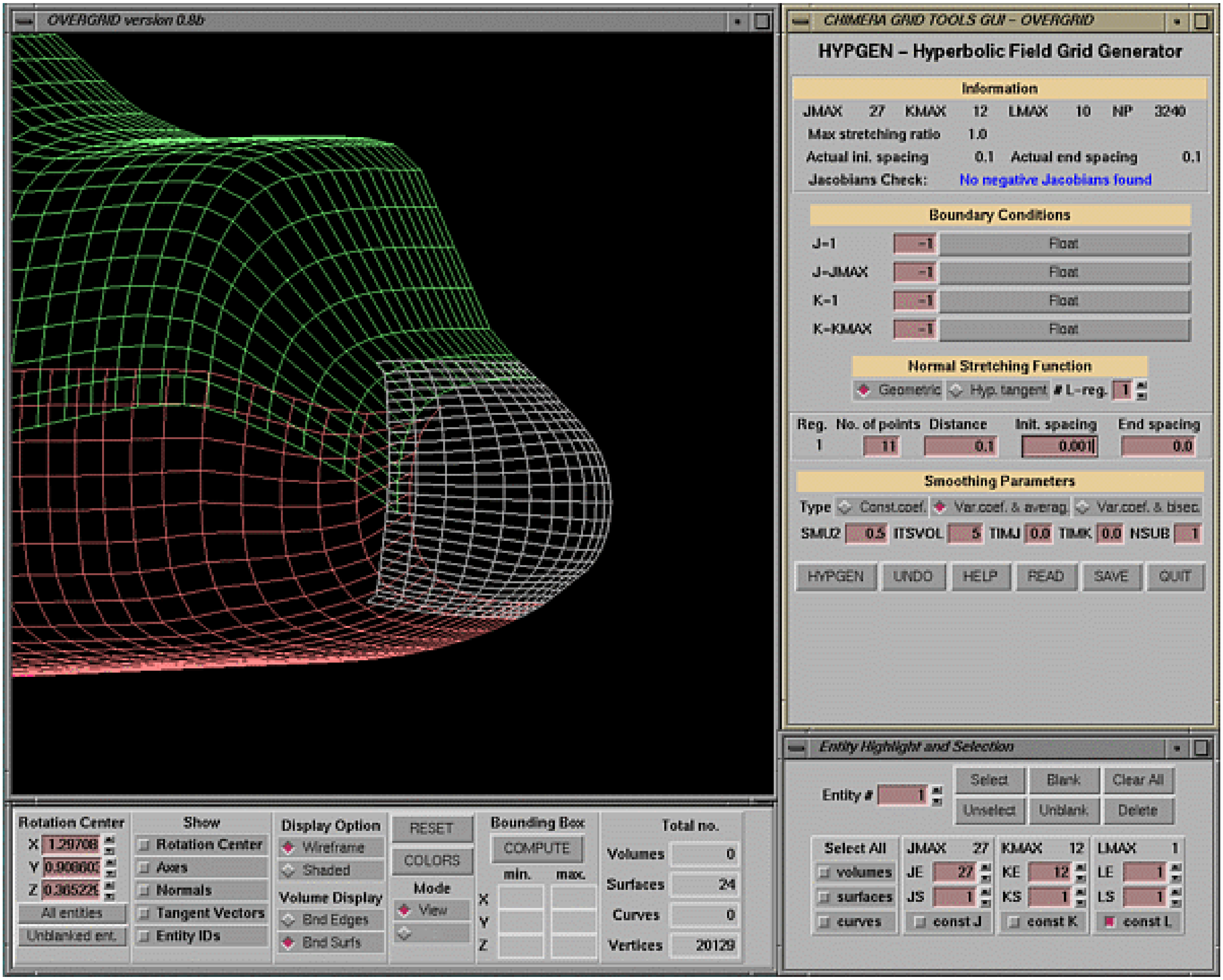

OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI – OVERGRID
SURGRD – Overset Surface Grid Generator
Total No. Show
Ref. Surfaces 28
Ref. Curves 36
Surface Grids 1
Auto-concatenate
Auto-project
Define Families
No. of defined families 0
Highlight: Family 0 Ref. surface 0 Ref. curve 0
Input Parameters for Surface Grid 1
Use hi-lighted family Family 0 Ini. Curve List: 1
Dir. + –
Add Remove Replace
Boundary Conditions
J=1 J=JMAX
Float Float
Normal Stretching Function
Specify Input
Number of points 11
Stretching Ratio
No. of nodes 1
Info:
max str. ratio = 1.106
Node J index Distance Init. spacing End spacing
1 1 0.15 0.01 0.0
Smoothing Parameters
SMU 0.5 TIM 0.0 ITSVOL 5
SURGRD UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 0
Select Blank Clear All
Unselect Unblank
Select All JMAX 0 KMAX 0 LMAX 0
volumes JE 0 KE 0 LE 0
surfaces JS 0 KS 0 LS 0
curves const J const K const L
Rotation Center
X 1.29709
Y 0.908602
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 0
Surfaces 28
Curves 36
Vertices 17834

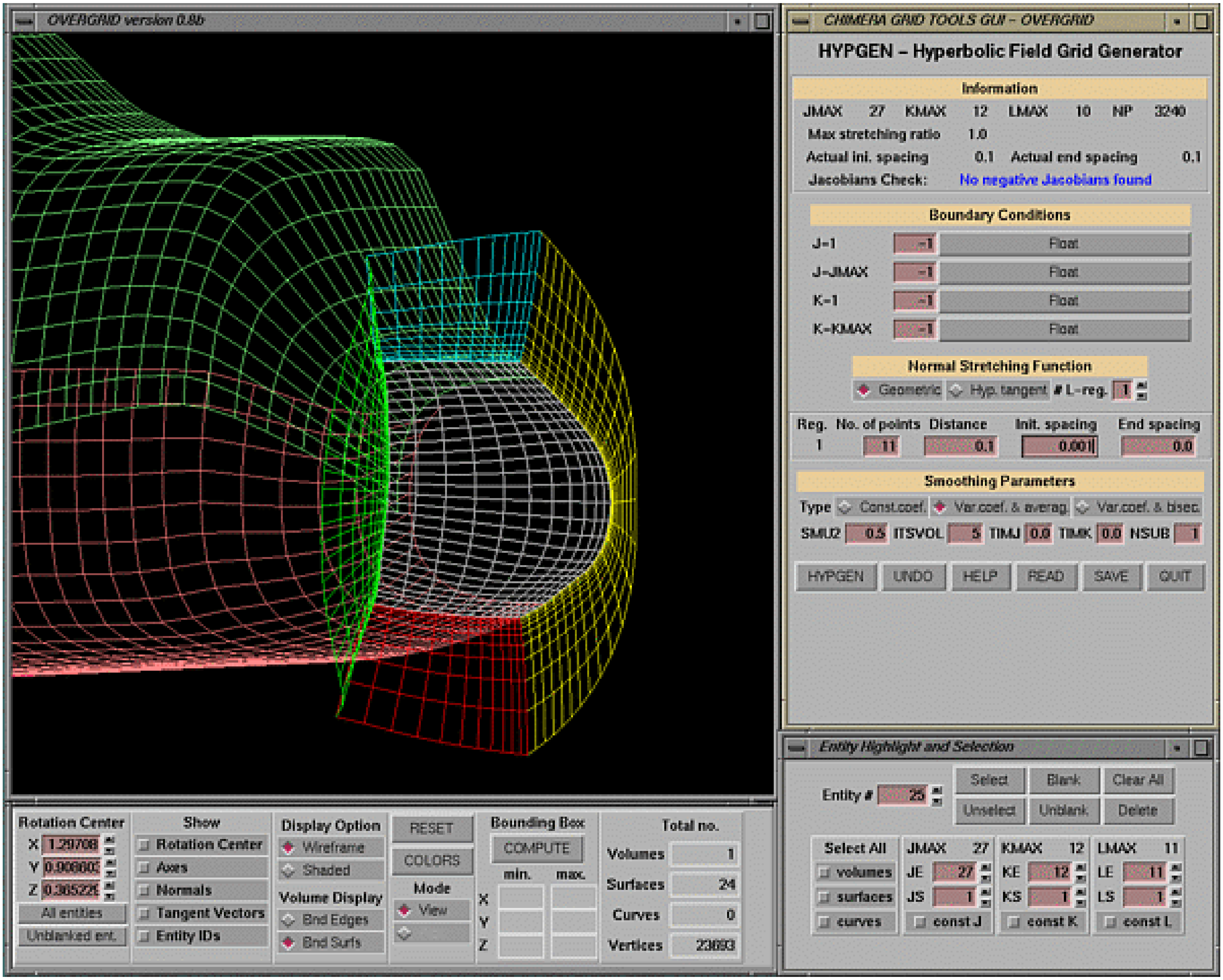

OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
SURGRD - Overset Surface Grid Generator
Total No.
Show
Ref. Surfaces
28
Ref. Curves
36
Surface Grids
1
Auto-concatenate
Auto-project
Define Families
No. of defined families 0
Highlight: Family 0 Ref. surface 0 Ref. curve 0
Input Parameters for Surface Grid 1
Use hi-lighted family
Family 0
Ini. Curve List: 1
Dir.
Add
Remove
Replace
Boundary Conditions
J=1
J=JMAX
Float
Float
1
1
Normal Stretching Function
Specify
Input
Number of points
11
Stretching Ratio
No. of nodes 1
Info:
max str. ratio = 1.106
Node
J index
Distance
Init. spacing
End spacing
1
1
0.15
0.01
0.0
Smoothing Parameters
SMU 0.5
TIM 0.0
ITSVOL 5
SURGRD
UNDO
HELP
READ
SAVE
QUIT
Entity Highlight and Selection
Entity # 0
Select
Blank
Clear All
Unselect
Unblank
Select All
volumes
surfaces
curves
JMAX 0
JE 0
JS 0
const J
KMAX 0
KE 0
KS 0
const K
LMAX 0
LE 0
LS 0
const L
Rotation Center
X 1.29709
Y 0.908803
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min.
max.
X
Y
Z
Total no.
Volumes 0
Surfaces 28
Curves 36
Vertices 17834

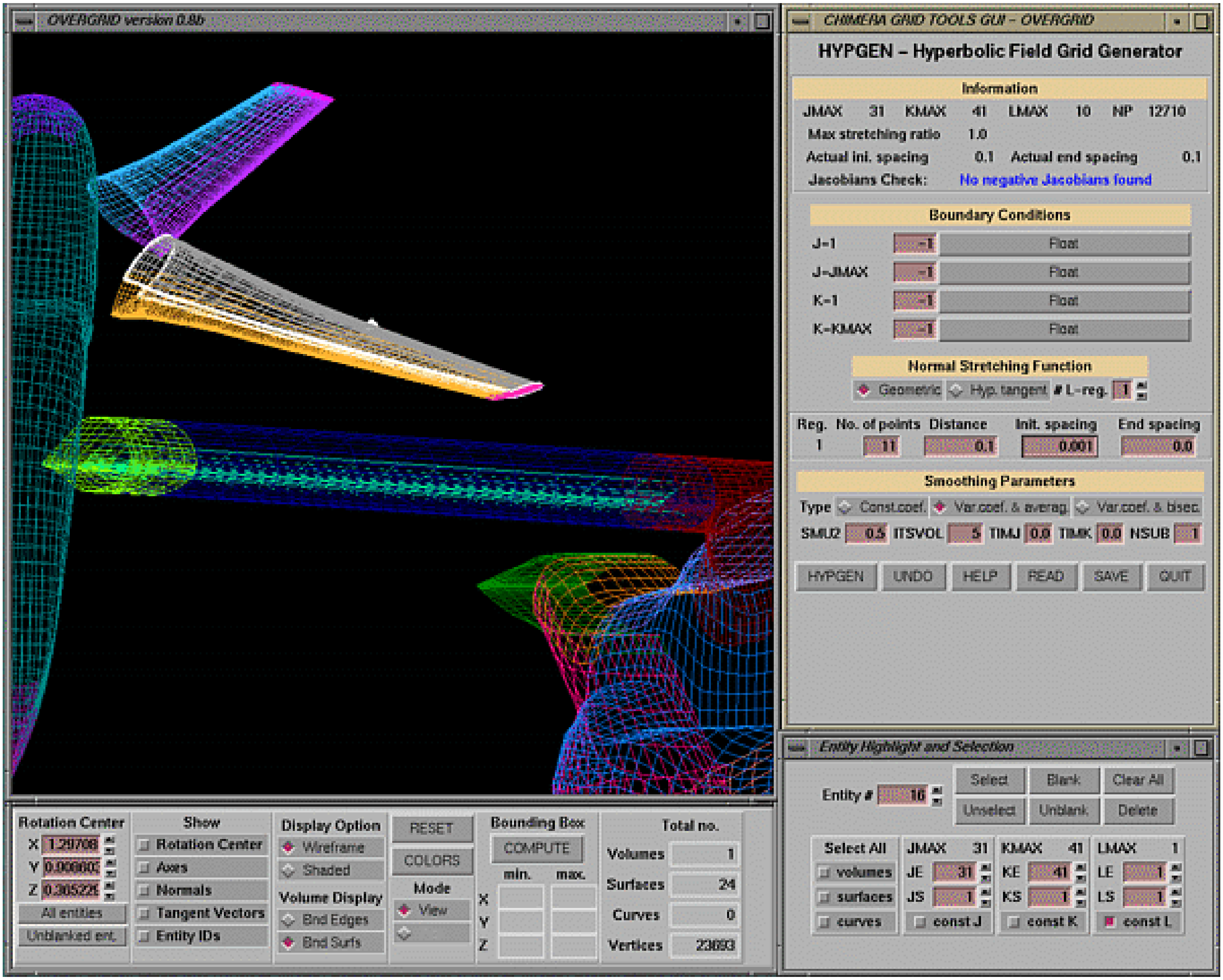

OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
SURGRD - Overset Surface Grid Generator
Total No.
Show
Ref. Surfaces 28
Ref. Curves 36
Surface Grids 1
Auto-concatenate
Auto-project
Define Families
No. of defined families 0
Highlight: Family 0 Ref. surface 0 Ref. curve 0
Input Parameters for Surface Grid 1
Use hi-lighted family Family 0 Ini. Curve List: 1
Dir. + -
Add Remove Replace
Boundary Conditions
J=1 J=JMAX
Float Float
Normal Stretching Function
Specify Input
Number of points 11
Stretching Ratio
No. of nodes 1
Info:
max str. ratio = 1.106
Node J index Distance Init. spacing End spacing
1 1 0.15 0.01 0.0
Smoothing Parameters
SMU 0.5 TIM 0.0 ITSVOL 5
SURGRD UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 0
Select Blank Clear All
Unselect Unblank
Select All
volumes
surfaces
curves
JMAX 0 KMAX 0 LMAX 0
JE 0 KE 0 LE 0
JS 0 KS 0 LS 0
const J const K const L
Rotation Center
X 1.29709
Y 0.900803
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 0
Surfaces 28
Curves 36
Vertices 17834

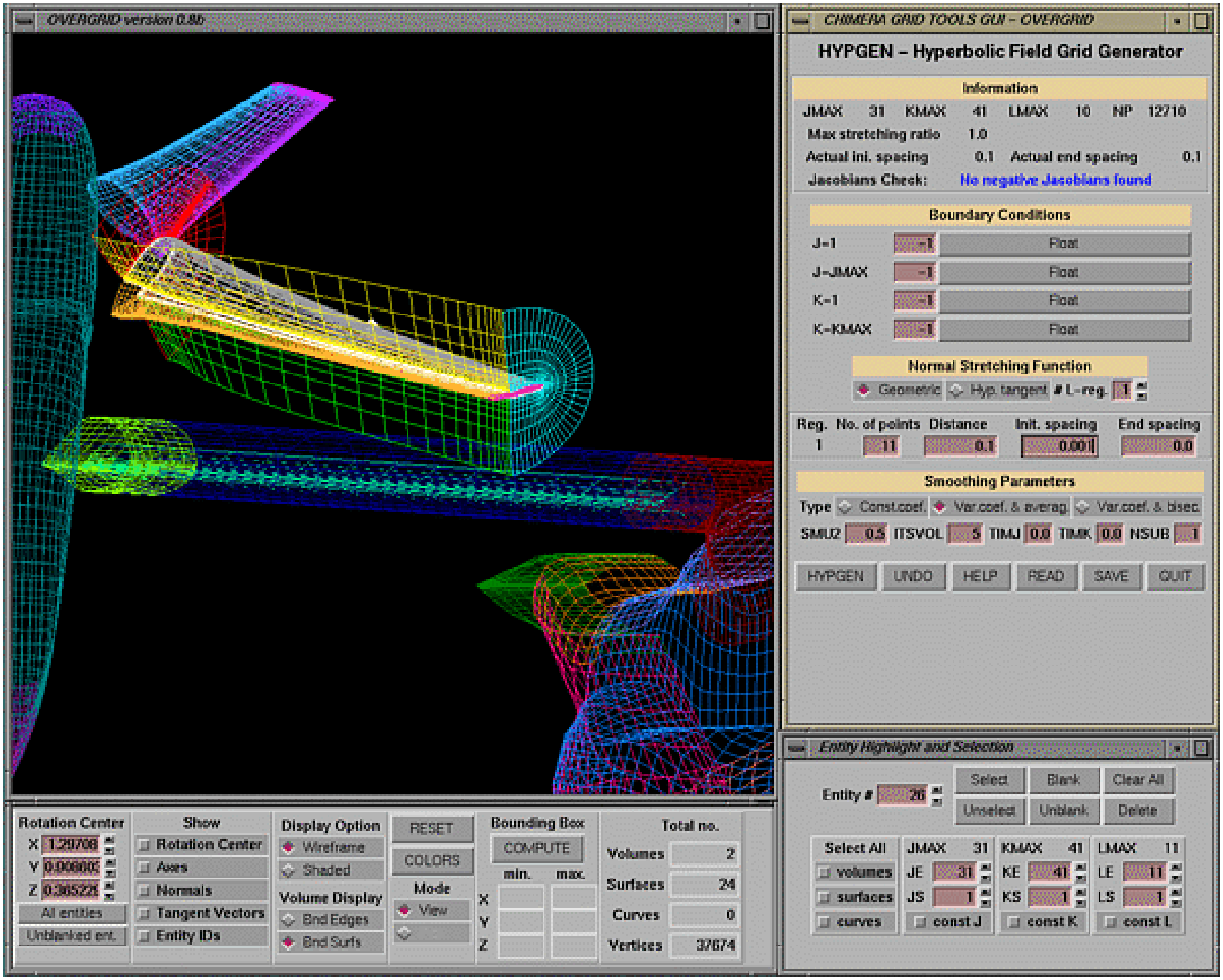

OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
SURGRD - Overset Surface Grid Generator
Total No.
Show
Ref. Surfaces
28
Ref. Curves
36
Surface Grids
41
Auto-concatenate
Auto-project
Define Families
No. of defined families 0
Highlight: Family 0
Ref. surface 0
Ref. curve 0
Input Parameters for Surface Grid 1
Use hi-lighted family
Family 0
Ini. Curve List: 1
Dir.
Add
Remove
Replace
Boundary Conditions
J=1
J=JMAX
Float
Float
1
1
Normal Stretching Function
Specify
Input
No. of nodes 3
Number of points 12
Info:
Stretching Ratio
max str. ratio = 1.244
Node
J index
Distance
Init. spacing
End spacing
1
1
0.08
0.004
0.0
2
13
0.13
0.004
0.0
3
-1
0.08
0.004
0.0
Smoothing Parameters
SMU 0.7
TIM 3.0
ITSVOL 2
SURGRD
UNDO
HELP
READ
SAVE
QUIT
Entity # 0
Select
Blank
Clear All
Unselect
Unblank
Select All
JMAX 0
KMAX 0
LMAX 0
volumes
JE 0
KE 0
LE 0
surfaces
JS 0
KS 0
LS 0
curves
const J
const K
const L
Rotation Center
X 1.29709
Y 0.908602
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min.
max.
X
Y
Z
Total no.
Volumes 0
Surfaces 28
Curves 36
Vertices 17834

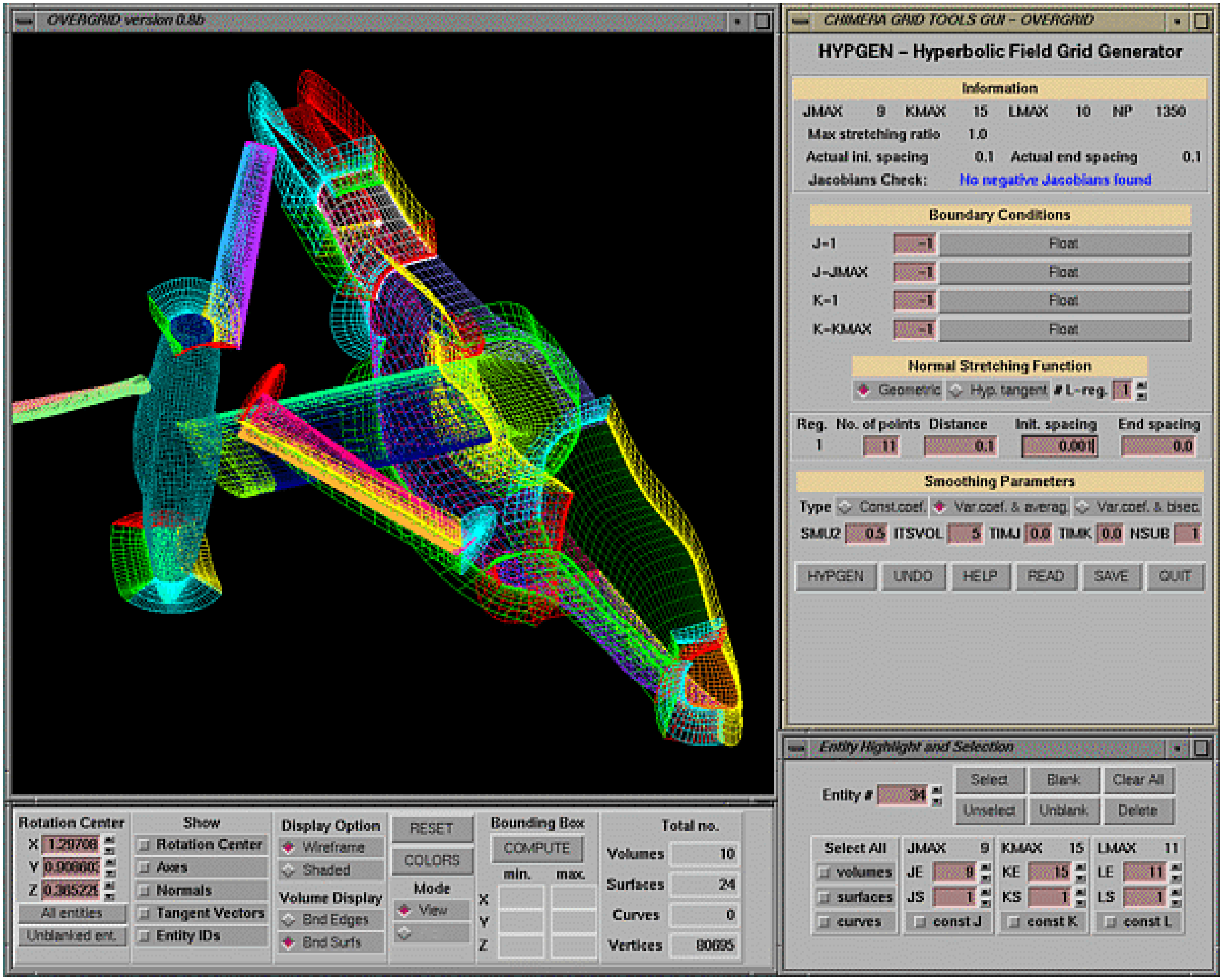

OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI – OVERGRID
SURGRD – Overset Surface Grid Generator
Total No. Show
Ref. Surfaces 28
Ref. Curves 36
Surface Grids 41
Auto-concatenate
Auto-project
Define Families
No. of defined families 0
Highlight: Family 0 Ref. surface 0 Ref. curve 0
Input Parameters for Surface Grid 1
Use hi-lighted family Family 0 Ini. Curve List: 1
Dir. + –
Add Remove Replace
Boundary Conditions
J=1 J=JMAX
Float Float
1 1
Normal Stretching Function
Specify Input
Number of points 12
Stretching Ratio
No. of nodes 3
Info:
max str. ratio = 1.244
Node J index Distance Init. spacing End spacing
1 1 0.08 0.004 0.0
2 13 0.13 0.004 0.0
3 -1 0.08 0.004 0.0
Smoothing Parameters
SMU 0.7 TIM 3.0 ITSVOL 2
SURGRD UNDO HELP READ SAVE QUIT
Entity # 0
Select Blank Clear All
Unselect Unblank
Select All
volumes
surfaces
curves
JMAX 0 KMAX 0 LMAX 0
JE 0 KE 0 LE 0
JS 0 KS 0 LS 0
const J const K const L
Rotation Center
X 1.29709
Y 0.908803
Z 0.36523
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 0
Surfaces 28
Curves 36
Vertices 17834

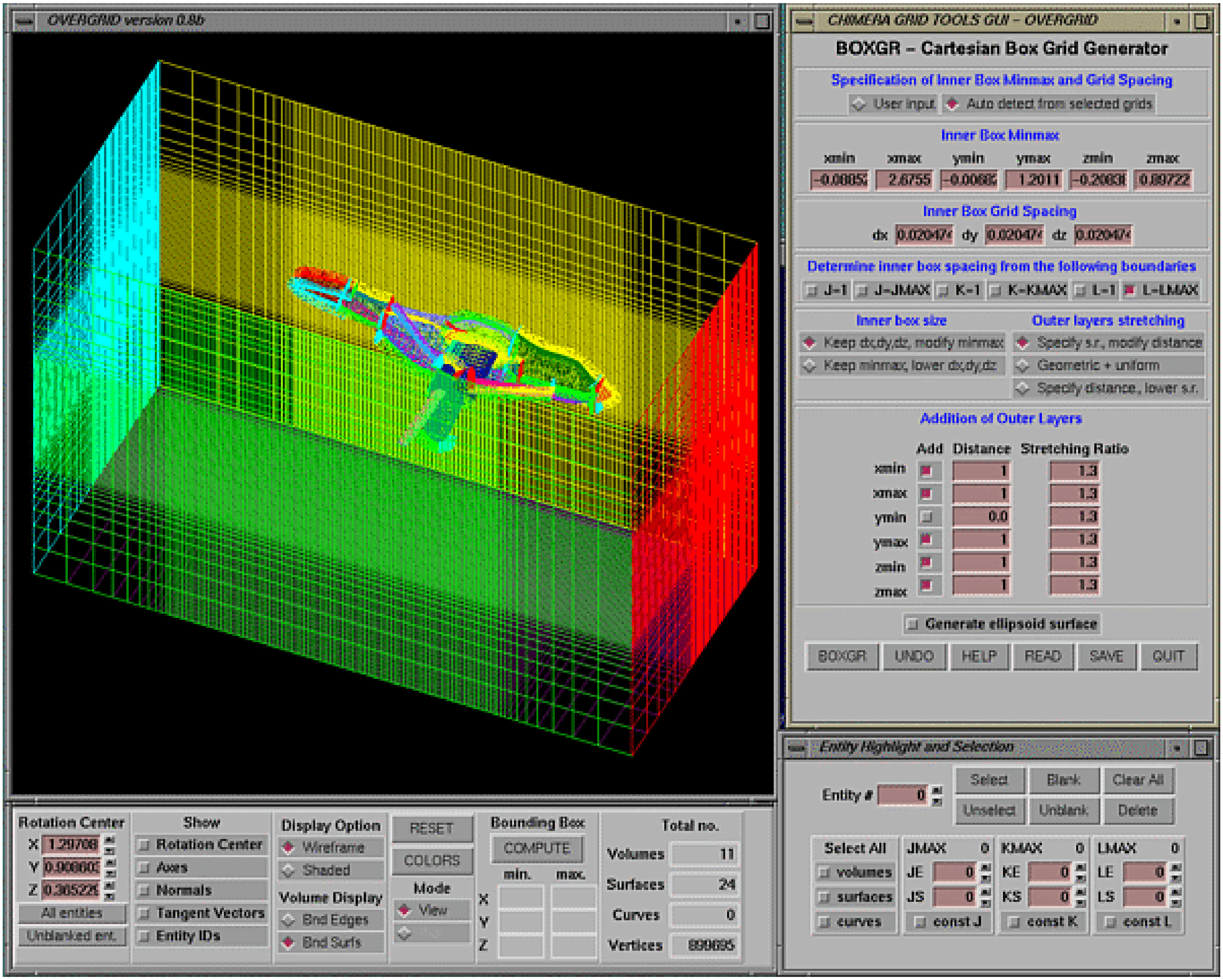

OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
HYPGEN - Hyperbolic Field Grid Generator
Information
JMAX 27 KMAX 12 LMAX 10 NP 3240
Max stretching ratio 1.0
Actual ini. spacing 0.1 Actual end spacing 0.1
Jacobians Check: No negative Jacobians found
Boundary Conditions
J=1 -1 Float
J=JMAX -1 Float
K=1 -1 Float
K=KMAX -1 Float
Normal Stretching Function
Geometric Hyp. tangent # L-reg. 1
Reg. No. of points Distance Init. spacing End spacing
1 11 0.1 0.001 0.0
Smoothing Parameters
Type Const.coef. Var.coef. & averag. Var.coef. & bisec.
SMU2 0.5 ITSVOL 5 TIMJ 0.0 TIMK 0.0 NSUB 1
HYPGEN UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 1
Select Blank Clear All
Unselect Unblank Delete
Select All JMAX 27 KMAX 12 LMAX 1
volumes JE 27 KE 12 LE 1
surfaces JS 1 KS 1 LS 1
curves const J const K const L
Rotation Center
X 1.29708
Y 0.908603
Z 0.365229
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 0
Surfaces 24
Curves 0
Vertices 20129

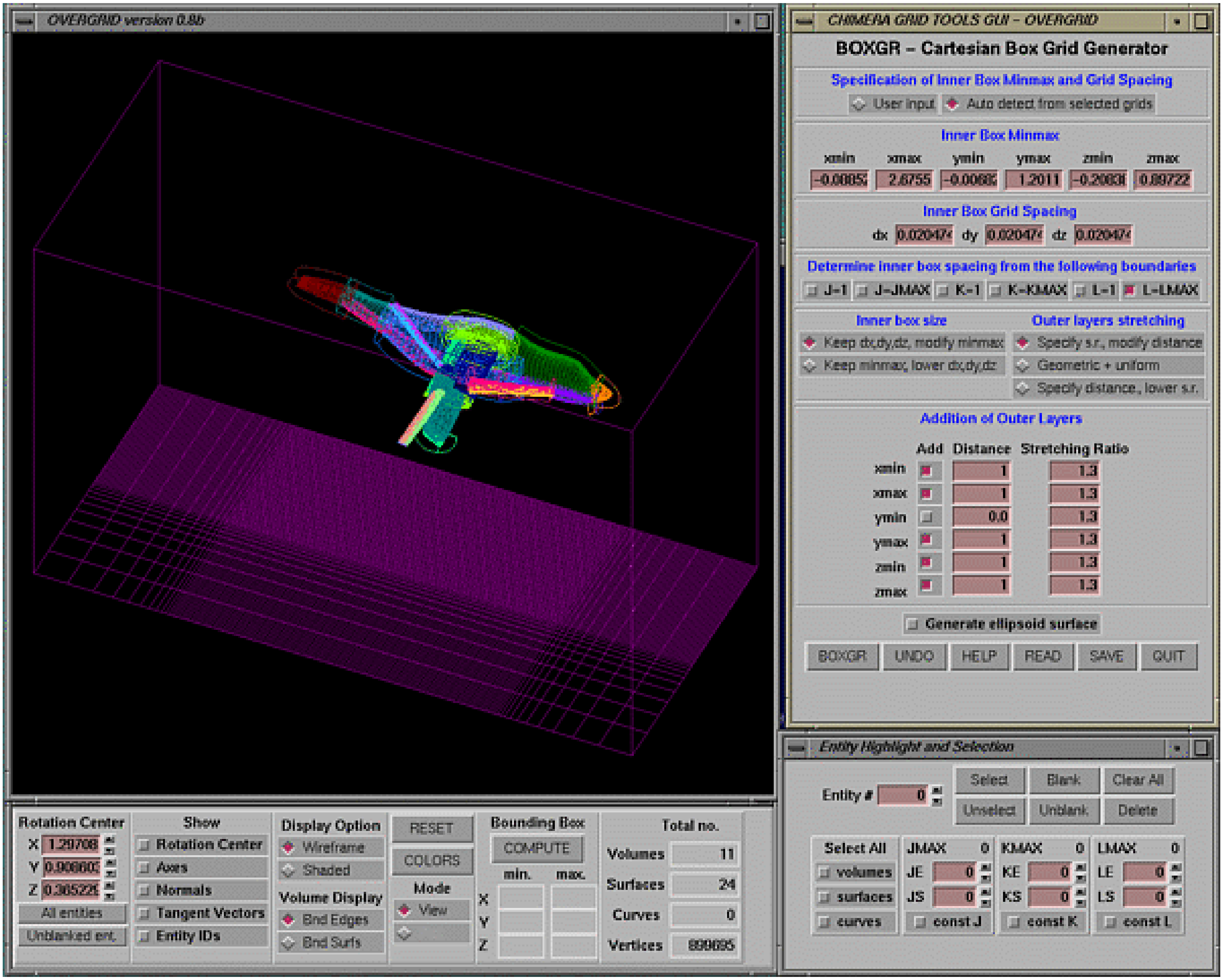

OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
HYPGEN - Hyperbolic Field Grid Generator
Information
JMAX 27 KMAX 12 LMAX 10 NP 3240
Max stretching ratio 1.0
Actual ini. spacing 0.1 Actual end spacing 0.1
Jacobians Check: No negative Jacobians found
Boundary Conditions
J=1 -1 Float
J=JMAX -1 Float
K=1 -1 Float
K=KMAX -1 Float
Normal Stretching Function
Geometric Hyp. tangent # L-reg. 1
Reg. No. of points Distance Init. spacing End spacing
1 11 0.1 0.001 0.0
Smoothing Parameters
Type Const.coef. Var.coef. & averag. Var.coef. & bisec.
SMU2 0.5 ITSVOL 5 TIMJ 0.0 TIMK 0.0 NSUB 1
HYPGEN UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 25
Select Blank Clear All
Unselect Unblank Delete
Select All
volumes
surfaces
curves
JMAX 27 KMAX 12 LMAX 11
JE 27 KE 12 LE 11
JS 1 KS 1 LS 1
const J const K const L
Rotation Center
X 1.29708
Y 0.906603
Z 0.365229
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 1
Surfaces 24
Curves 0
Vertices 23693


OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
HYPGEN - Hyperbolic Field Grid Generator
Information
JMAX 31 KMAX 41 LMAX 10 NP 12710
Max stretching ratio 1.0
Actual ini. spacing 0.1 Actual end spacing 0.1
Jacobians Check: No negative Jacobians found
Boundary Conditions
J=1 -1 Float
J=JMAX -1 Float
K=1 -1 Float
K=KMAX -1 Float
Normal Stretching Function
Geometric Hyp. tangent # L-reg. 1
Reg. No. of points Distance Init. spacing End spacing
1 11 0.1 0.001 0.0
Smoothing Parameters
Type Const.coef. Var.coef. & averag. Var.coef. & bisec.
SMU2 0.5 ITSVOL 5 TIMJ 0.0 TIMK 0.0 NSUB 1
HYPGEN UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 16
Select Blank Clear All
Unselect Unblank Delete
Select All
volumes
surfaces
curves
JMAX 31 JE 31 JS 1 const J
KMAX 41 KE 41 KS 1 const K
LMAX 1 LE 1 LS 1 const L
Rotation Center
X 1.29708
Y 0.00000
Z 0.305226
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 1
Surfaces 24
Curves 0
Vertices 23693


OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
HYPGEN - Hyperbolic Field Grid Generator
Information
JMAX 31 KMAX 41 LMAX 10 NP 12710
Max stretching ratio 1.0
Actual ini. spacing 0.1 Actual end spacing 0.1
Jacobians Check: No negative Jacobians found
Boundary Conditions
J=1 -1 Float
J=JMAX -1 Float
K=1 -1 Float
K=KMAX -1 Float
Normal Stretching Function
Geometric Hyp. tangent # L-reg. 1
Reg. No. of points Distance Init. spacing End spacing
1 11 0.1 0.001 0.0
Smoothing Parameters
Type Const.coef. Var.coef. & averag. Var.coef. & bisec.
SMU2 0.5 ITSVOL 5 TIMJ 0.0 TIMK 0.0 NSUB 1
HYPGEN UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 26
Select Blank Clear All
Unselect Unblank Delete
Select All JMAX 31 KMAX 41 LMAX 11
volumes JE 31 KE 41 LE 11
surfaces JS 1 KS 1 LS 1
curves const J const K const L
Rotation Center
X 1.29708
Y 0.906003
Z 0.365225
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 2
Surfaces 24
Curves 0
Vertices 37674


OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI – OVERGRID
HYPGEN – Hyperbolic Field Grid Generator
Information
JMAX 9 KMAX 15 LMAX 10 NP 1350
Max stretching ratio 1.0
Actual ini. spacing 0.1 Actual end spacing 0.1
Jacobians Check: No negative Jacobians found
Boundary Conditions
J=1 -1 Float
J=JMAX -1 Float
K=1 -1 Float
K=KMAX -1 Float
Normal Stretching Function
Geometric Hyp. tangent # L-reg. 1
Reg. No. of points Distance Init. spacing End spacing
1 11 0.1 0.001 0.0
Smoothing Parameters
Type Const.coef. Var.coef. & averag. Var.coef. & bisec.
SMU2 0.5 ITSVOL 5 TIMJ 0.0 TIMK 0.0 NSUB 1
HYPGEN UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 34
Select Blank Clear All
Unselect Unblank Delete
Select All
volumes
surfaces
curves
JMAX 9 JE 9 JS 1 const J
KMAX 15 KE 15 KS 1 const K
LMAX 11 LE 11 LS 1 const L
Rotation Center
X 1.29708
Y 0.906603
Z 0.305229
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 10
Surfaces 24
Curves 0
Vertices 80695


OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
BOXGR - Cartesian Box Grid Generator
Specification of Inner Box Minmax and Grid Spacing
User input
Auto detect from selected grids
Inner Box Minmax
xmin xmax ymin ymax zmin zmax
2.6755
1.2011
Inner Box Grid Spacing
dx dy dz
Determine inner box spacing from the following boundaries
J=1 J=JMAX K=1 K=KMAX L=1 L=LMAX
Inner box size
Keep dx,dy,dz, modify minmax
Keep minmax, lower dx,dy,dz
Outer layers stretching
Specify s.r., modify distance
Geometric + uniform
Specify distance, lower s.r.
Addition of Outer Layers
Add Distance Stretching Ratio
xmin 1 1.3
xmax 1 1.3
ymin 0.0 1.3
ymax 1 1.3
zmin 1 1.3
zmax 1 1.3
Generate ellipsoid surface
BOXGR UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 0
Select Blank Clear All
Unselect Unblank Delete
Select All
volumes
surfaces
curves
JMAX 0 JE 0 JS 0 const J
KMAX 0 KE 0 KS 0 const K
LMAX 0 LE 0 LS 0 const L
Rotation Center
X 1.29708
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 11
Surfaces 24
Curves 0
Vertices 899695


OVERGRID version 0.8b
CHIMERA GRID TOOLS GUI - OVERGRID
BOXGR - Cartesian Box Grid Generator
Specification of Inner Box Minmax and Grid Spacing
User input
Auto detect from selected grids
Inner Box Minmax
xmin xmax ymin ymax zmin zmax
2.6755
1.2011
0.09722
Inner Box Grid Spacing
dx dy dz
Determine inner box spacing from the following boundaries
J=1 J=JMAX K=1 K=KMAX L=1 L=LMAX
Inner box size
Outer layers stretching
Keep dx,dy,dz, modify minmax
Keep minmax, lower dx,dy,dz
Specify s.r., modify distance
Geometric + uniform
Specify distance, lower s.r.
Addition of Outer Layers
Add Distance Stretching Ratio
xmin 1 1.3
xmax 1 1.3
ymin 0.0 1.3
ymax 1 1.3
zmin 1 1.3
zmax 1 1.3
Generate ellipsoid surface
BOXGR UNDO HELP READ SAVE QUIT
Entity Highlight and Selection
Entity # 0
Select Blank Clear All
Unselect Unblank Delete
Select All
volumes
surfaces
curves
JMAX 0
JE 0
JS 0
const J
KMAX 0
KE 0
KS 0
const K
LMAX 0
LE 0
LS 0
const L
Rotation Center
X 1.29708
All entities
Unblanked ent.
Show
Rotation Center
Axes
Normals
Tangent Vectors
Entity IDs
Display Option
Wireframe
Shaded
Volume Display
Bnd Edges
Bnd Surfs
RESET
COLORS
Mode
View
Bounding Box
COMPUTE
min. max.
X
Y
Z
Total no.
Volumes 11
Surfaces 24
Curves 0
Vertices 899695

New Cray Metacenter Designed With Researchers in Mind

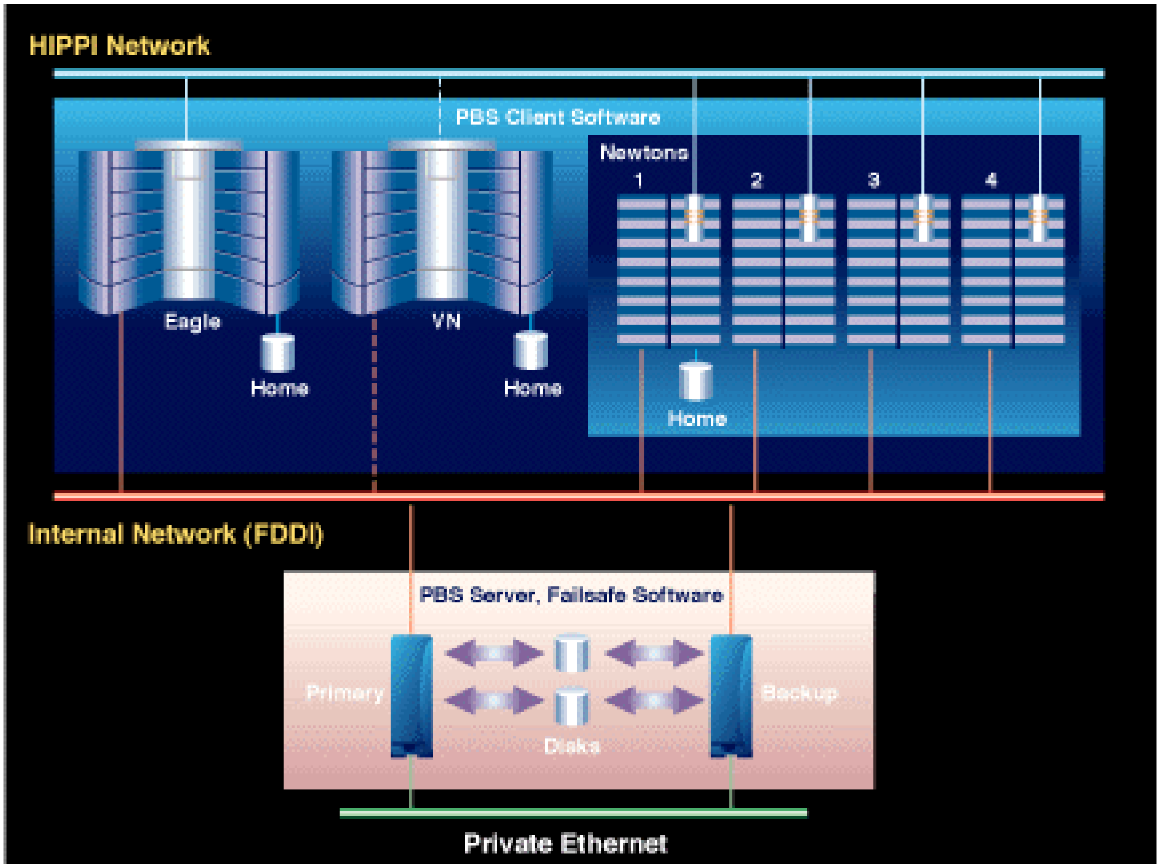
by Jill Dunbar

After every round of financial belt-tightening at NASA, additional computing resources seem to sink lower on the priority list. But like anyone who's fallen on hard times and makes the most of what they've got, the NAS Systems Division staff has come up with a creative method for squeezing the most computing cycles out of the resources at hand. To help meet the ever-increasing clamor for these resources, a new Cray metacenter has been developed to connect five Cray Research supercomputers funded by the Aeronautical Consolidated Supercomputing Facility (ACSF).

For more than a year, Cray systems housed at the NAS Facility have been overloaded. "Too many users chasing too few resources," is how Jim Weilmuenster, one of those users, puts it. The demand for computing time outstrips supply to such an extent that Weilmuenster, who does aerothermodynamics analysis for such projects as the Lockheed Martin X-33 and the Mars Pathfinder series, is skeptical about the metacenter's ability to solve his resource problems.

Sarita Wood, who developed the software underlying the metacenter, acknowledges that the new arrangement may not be the answer for everyone, but believes that it will help many of the 300-plus researchers who use the five ACSF supercomputers. These include Eagle, a CRAY

C90, and the Newton cluster, four CRAY J90s of varying sizes.



The Cray metacenter at the NAS Facility will initially include Eagle (a CRAY C90) and Newtons 1, 2, 3, and 4 (a cluster of CRAY J90s)all part of NASA's Aeronautical Consolidated Supercomputing Facility. Von Neumann (the NAS Systems Division's CRAY C90) will be added to the metacenter in October. Using the Portable Batch System (PBS) with Silicon Graphics Inc.'s FailSafe software provides the ability for the PBS scheduler to continue to run new jobs and for users to continue to submit jobs and get status when any of the supercomputers fail. Graphic by Gail Felchle.

All users with accounts on Eagle or the Newtons will have access to the metacenter. (The NAS Facility's CRAY C90, Von Neumann, will be added to the metacenter complex next October.)

The main idea behind the metacenter is to balance the workload based on memory and CPU requirements and fill unused cycles on all systems -- in particular the Newtons, which usually carry a lighter load than Eagle. This should improve total throughput for each system, as well as the average turnaround time for users' jobs, Wood says.

Improved Flexibility, Reliability, Turnaround

Back in November, when Wood and team members Nick Cardo, James Jones, and Alan Powers started kicking around the metacenter design, three items topped their list of goals: quicker job turnaround, near-foolproof reliability, and added flexibility for users. Wood's personal motivation is to help users make the most of the resources in any way she can (*see sidebar, below*).

An ongoing issue for both users and NAS's HSP support staff has been system reliability. "Right now, if the server for the Newton cluster [currently Newton1, shown in the system configuration at right] fails, jobs running on the other three Newtons have to be rerun from the beginning if they complete while the server is down," Wood explains.

The team is improving reliability by incorporating two dedicated Silicon Graphics (SGI) Origin2000 servers and IRIS FailSafe software into the metacenter configuration. The FailSafe server has no single point of failure -- barring a power outage -- and backup hardware automatically takes over in the event of failure. With this configuration, users can submit jobs to run on one of the Crays even when that system is down, and jobs already submitted can continue to run.

Briefly, here's how the metacenter works: Users log into their accounts on either Eagle or Newton1 as usual. They can then submit a script to specify certain job requirements, such as how much memory and how many CPUs are needed. The Portable Batch System (PBS) server keeps the job in a database. A scheduler determines where the job can run and then sends it to the first available system that meets those requirements. Job status is tracked to completion.

Enhancements to PBS allow users to choose an architecture type (C90 or J90) or a particular host if they wish. However, by letting jobs run on *any* of the five systems within the metacenter, users increase their chances for quicker job turnaround.

Complexity vs. Flexibility

Weilmuenster still worries that the metacenter is "adding another level of complexity to an already complex situation." Wood acknowledges the challenges -- particularly in ensuring that a single job is portable across all platforms supported within the metacenter. But, she says, other kinds of difficulties will diminish. For example, "Users will no longer need to check the load on various systems in an attempt to pick the machine that will turn the job around the fastest."

Handling architectural differences between the C90 and the J90s has also been tricky. Because the J90 processors are slower than the C90s, Wood modified PBS to normalize all time values to C90-equivalent hours. All jobs will be specified in C90 time, and if a job happens to run on a J90, the time will increase proportionally to get the same amount of work done on either system. Current Newton users must do a simple calculation to convert J90 hours to normalized C90 CPU hours. (*See "Making the Most of the New Cray Metacenter" for more detailed user information.*) They can check the elapsed time (in C90 equivalents) on a job at any point during the run.

It May -- or May Not -- Help 'Savvy' Users

Weilmuenster says that goals like load balancing, reliability, and flexibility are all well and good in theory, but are unlikely to help in practice. "Some pretty savvy people run on those machines. They already know what to do to make the best use of resources," he says. Weilmuenster may be in a special position, however: he has accounts on both Eagle and Von Neumann and can decide for himself the best way to run his jobs. Weilmuenster admits, though, that there's not much even *he* can do about bottlenecks when there are lots of users on both systems.

Just before Christmas, for example (when many researchers were busy preparing presentations for the Reno AIAA conference), turnaround took four days for Scott Lawrence, whose work involves the supersonic transport wing/body design for NASA's High Speed Research Program. "It's frustrating as hell, if I have a deadline," Lawrence says. "I think the metacenter is probably something we need."

Delivery of the Origin2000s was expected at the end of May, and final testing is planned for June. That means the metacenter should be available to users by mid-summer -- just as they are realizing that there are only a couple of months left to consume their remaining time on the systems.

The Cray metacenter design team is cautious but optimistic about reaching its goals. Improvements in turnaround time will depend on the types of jobs running at any given moment, Wood explains. But if throughput and machine usage are less than ideal, system administrators can still adjust the job mix on a given machine manually. "The hope," she says, "is to keep all machines busy all the time, and to get larger jobs through the system without penalizing users."

She puts her money where her mouth is

If Cray user Scott Lawrence could wave a magic wand and change just one thing about the NAS Facility's supercomputing environment, what would it be?

"I'm sort of a pathological case, probably," Lawrence muses. "I tend to run a lot of half-hour jobs. Not eight-hour jobs, not five-minute jobs. I'd like to be able to get a 30-minute 64-megaword job back in half a day." Lawrence, a researcher in the Applied Aerodynamics Computation Branch at Ames Research Center, has sometimes waited a week to get a job of this size back.

When this is relayed to Cray metacenter developer Sarita Wood-- who designed the metacenter with users in mind-- she doesn't waste a second. Picking up her phone, she dials Lawrence to let him know about a queue that already does just what he wants. A highbatch queue exists specifically to get 30-minute jobs through the system faster. Lawrence, who was unaware of the queue, is pleased with Wood's immediate response.

"At 64 megawordsa quarter of Eagle's memoryit's going to be slow-going anyway," Wood acknowledges. But her philosophy is that every little bit may help.

NASnews
Main Menu
May–June 1998 • Volume 3, Number 3

Taking Computational Fluid Dynamics Into the Next Century: Origin2000s Grab Users' Attention

by Ayse Sercan

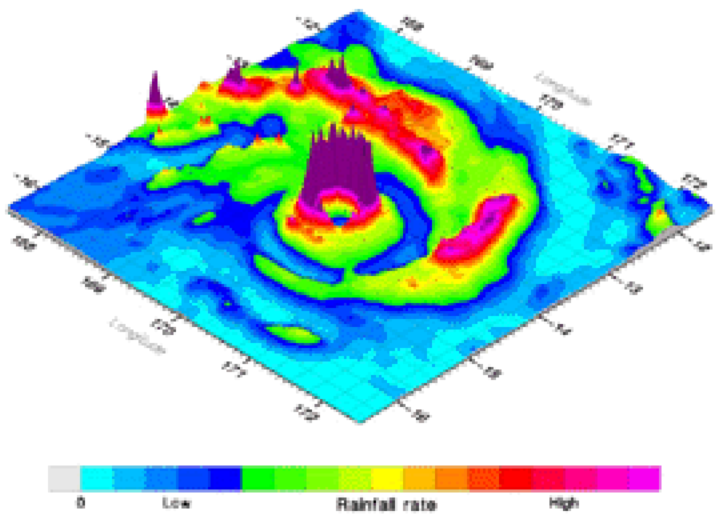
Much of the time, traffic on the CRAY C90 supercomputers at the NAS Facility rivals commuter bottlenecks on California's 101 freeway, just a stone's throw away from NASA Ames Research Center. Many users, however, are discovering a quicker, less crowded route to their scientific destinations: five Silicon Graphics Inc. (SGI) Origin2000 machines, known as Turing, JimPf0 and JimPf1, Hopper, Evelynand the newest arrivals, Heisenberg and Steger (*see endnote*).

Some researchers are drawn to the Origin2000s because the machines' parallel computing architecture is perfectly suited to solving their particular scientific or engineering problems. Others are simply finding that queues for CPU time are shorter or that code written for serial machines runs faster once it's been parallelized.

Here, three groups of outside investigators using the SGI Origin2000s explain what their research is about, how they've adapted their legacy code to run on parallel machines, and how well the Origin2000s are serving their needs. All seem to agree that the new machines can handle big problems with unexpected swiftness.

Observing the Earth

Measuring cloud cover, rainfall, and heat circulation in the atmosphere is one important goal of NASA's Earth Science enterprise. A series of satellite-borne instruments developed at Langley Research Center as part of the Clouds and the Earth's Radiant Energy System (CERES) experiment will gather such measurements over a 15-year period, allowing atmospheric scientists to study the connection between Earth's overall absorption of solar radiation and climate change.



NASA's Data Assimilation Office (DAO) is using the Silicon Graphics Inc. Origin2000s at the NAS Facility to turn raw measurements from Goddard Space Flight Center's Tropical Rainfall Measuring Mission (TRMM) satellite into useable data. This rendering of rainfall rate during Cyclone Susan (last January) shows some of that data after it has been transformed. Graphic courtesy of James G. Stobie. (See

[QuickTime movie of Cyclone Susan.)]

Transforming raw measurements into meaningful data, as in the image shown at upper right, involves complex retrieval methods. Key ingredients for the CERES cloud retrieval algorithm and other Earth Observing System (EOS) instruments include supplementary atmospheric information from Goddard Space Flight Center's Data Assimilation Office (DAO). This information is based on massive amounts of data from a wide variety of sources around the world. These complex algorithms demand the most powerful computing enginesexplaining why DAO shares ownership of several of the Origin2000 systems housed at NAS.

Three space missions carry CERES instruments: the Tropical Rainfall Measuring Mission (TRMM) satellite, launched last fall; the EOS-AM satellite, due to be launched later this year; and EOS-PM, scheduled for launch in 2000. TRMM, a joint mission between NASA and the National Space Development Agency of Japan, is typical of the CERES missions in that it will gather vast amounts of datain this case, visible and infrared measurements of energy rising from the Earth's surfacethat can be manipulated more quickly in parallel. Like other Origin2000 users, however, the DAO has found that parallelizing legacy codes originally written for the C90 is a challenge. (See *NAS News* September-October '97

"Right now, our focus is on producing global datasets of the Earth's atmosphere to help the TRMM," explains atmospheric scientist James G. Stobie, DAO's deputy chief. In the long term, the DAO will also assimilate new data generated by the instruments aboard these and other missions to create even more accurate records of the Earth's atmospherework for which the Origin2000s will be indispensable. "Assuming we can effectively parallelize our code, we should be able to get a much more accurate picture of the atmosphere than previously available," says Stobie. "There's a direct correlation between the quality of our product and the amount of computing power we have access to."

Untangling Wake Vortices at Boeing

Led by principal engineer Susan X. Ying, researchers at The Boeing Company in Long Beach will soon be using the Reynolds-averaged Navier-Stokes method to study the hazardous wake vortices left by transport airplanes.

Boeing principal engineer Susan X. Ying is studying the characteristics of the wake vortices of a transport airplane using the Reynolds-Averaged Navier-Stokes approach on the Origin2000s. This photograph shows the complicated flows in the dust cloud lifted into the wake of a transport airplane in high-lift configuration. Ying's simulation of this scenario ran four times faster on Turing (one of five SGI Origin2000 systems at the NAS Facility) than on a Cray Research CRAY C90.

As an airplane's wings generate lift, they shed vortices of air, and these vortices increase in strength in rough proportion to the aircraft's weight. To ensure safety, regulations demand that aircraft maintain a specific separation distance based on weighta constraint that seriously limits traffic flow at the nation's increasingly congested airports.

In the past, researchers have approximated wake vortices using large eddy simulations, direct numerical simulations, and vortex particle methods. Ying's team, however, will be studying the wake flow near aircraft surfaces by solving the Reynolds-averaged Navier-Stokes equations, using a program called OVERFLOW. This way of examining wake vortices is more accurate, but has never been attempted on an entire airplane model because of the staggering amount of computer power required. On a CRAY C90, for example, hundreds of CPU hours per case would be required to generate the tens of millions of points on the model's computational mesh. In OVERFLOW tests using Turing, however, Ying's group has already reduced this time to a handful of hours.

"We initially wanted to use Von Neumann [the NAS C90] for the computation, but getting time on the machine was very difficult," Ying explains. "Then NAS suggested that we try using the Origins." This was a risky move, since the Origin2000s were an untried platform for this kind of computation. But NAS physicist Jim Taft had created a new version of OVERFLOW using multilevel parallelism (MLP), a programming technique that capitalizes on the Origin2000 architecture, with many processors accessing a common memory. And when Ying gave Turing a dataset using a 35-million-point grid to represent a transport aircraft in high-lift configuration, she got results that she calls "astounding."

"When we ran NAS's MLP version of the code on Turing, it was four times faster than on Von Neumann," says Ying. "The code performance on the Origin2000 is very encouraging." Using 64 of Turing's processors, Ying's problem took only 45 seconds per iterationcompared to 170 seconds on the C90 using 16 processors and multi-tasking. Even running with only 32 processors, Turing took only 90 seconds per iteration to solve the problem.

"We were very fortunate to be able to work with Jim Taft on thishe's a genius," says Ying. Not only will Boeing's team be able to take advantage of the greater availability of the Origin2000s, she notes, but they will also be able to run two jobs in one day, compared to the one they could complete on the C90.

Optimizing the Supersonic Transport

Like many Origin2000 users, researcher Robert Biedron, in the Fluid Mechanics and Acoustics Division of NASA's Langley Research Center, has spent much of his time lately studying what happens when code designed to be executed on a single processor is distributed to many processors. A powerful program developed at Langley called CFL3D simulates air flow around an aircraft by solving the Navier-Stokes equations in three dimensions. Biedron wants to use CFL3D to model chunks of the envisioned hypersonic passenger jet known as the High Speed Civil Transport (HSCT)a computationally intensive job that would require unreasonable amounts of CPU time on the C90. "What I've been doing to date ... is making sure CFL3D can run reasonably well on the Origin," says Biedron. (Tests of CFL3D on an Origin2000 by NAS researcher Tom Faulkner last year have already produced promising results; see *NAS News* November-December 1997, page 3.)

Robert Biedron's aerodynamics research at Langley Research Center will help U.S. aircraft manufacturers effectively design the next-generation supersonic transport. Biedron is currently doing an aerodynamic analysis, taking advantage of the SGI Origin2000's parallel processing for CPU-intensive computations.

NASA has been studying ways to build an economically viable, environmentally acceptable HSCT since the early 1990s. Biedron's team is gearing up for the multidisciplinary design optimization (MDO) phase of the project. This process is aimed at developing a framework to be used by U.S. manufacturers to build a real-world aircraft. Using CFL3D, researchers can study the effects of changes in this framework without building actual models. "The initial optimization problem will be for a somewhat simplified configuration, but it is a real world optimization in the sense that the design will be driven by several disciplines: structures, aerodynamics, and performance," explains Biedron.

During the MDO phase, investigators must determine not only whether a particular aircraft design meets mission requirements, but how it behaves under stress and deformation. As a result, a mind-numbing volume of aerodynamic calculations must be performed. Because these calculations are by far the most CPU-intensive part of the MDO process, Biedron's team decided early to run them on a parallel platformpreferably the Origin2000s at NAS. And in recent tests, he's obtained encouraging results. When Biedron split an initial grid consisting of some 600,000 points into 24 blocks and analyzed them on 24 processors, performance improved by a factor of approximately 20. A job that would have required 3.4 hours on a single processor was finished in only 10 minutesa time savings that Biedron says is crucial to using computational fluid dynamics in the MDO process.

Tell Us Your Story

If you'd like to see your research using NAS Facility systems featured in a future issue of *NAS News*, send email to editor Jill Dunbar.

Two New Origin2000 Systems Arrive

Two more Silicon Graphics Inc. Origin2000 systems arrived at the NAS Facility on April 2. The machines, named Heisenberg and Steger, each include 128 processors, configured with .6 terabytes of disk space and 32 gigabytes of main memoryfour times the memory of the NAS CRAY C90. The two machines have enough memory and CPU capacity to solve problems involving 100 million points or more.

The high degree of parallelism offered by these systems promises to shorten execution time for some jobs by a factor of four over the C90. NAS Systems Division staff members are busy parallelizing popular CFD codes originally written for vector machines such as the C90 (*see next article*). For more information on the new systems, see the NAS Facility's Origin2000 website.

NASnews
Main Menu
May–June 1998 • Volume 3, Number 3

Making the Most of the New Cray Metacenter

by George Myers

The Cray metacenter, designed for the Aeronautical Consolidated Supercomputer Facility's CRAY C90, Eagle, and the CRAY J90 cluster, Newton, will allow all users with accounts on these systems to submit jobs based on their resource requirements. The benefit to individual users may be faster turnaround for jobs that can run on either platform, and the throughput and turnaround for the user base as a whole should improve because resource usage and load are more evenly distributed across the entire metacenter (*see Cray Metacenter Designed for Users*).

The metacenter is implemented using PBS, the Portable Batch System. The same familiar PBS commands (such as *qsub*, *qstat*, *qdel*) are used to submit, get status, and manage jobs. An added benefit is that a job may be submitted to the metacenter queues regardless of whether the target system is currently available. This article discusses several PBS features that handle the differences among the systems and make the metacenter easier to use.

Specify Resources Accurately

The NAS scientific consultants always recommend that resource requirements be specified as accurately as possible so that no system resources are wasted by overstated requests. With the metacenter, this becomes even more important. Why? The metacenter's "limit queues" describe the resources available for each queue on each system. Your job's resource requirements are checked against these queues to determine where the job might fit. All resources are considered: memory, scratch file space on the $BIGDIR and $FASTDIR file systems, number of CPUs, CPU time, and wall-clock time.

To compensate for the difference between C90 and J90 CPUs, all CPU time is measured in C90 hours, with a J90 being equivalent to a factor of 4.32. If, for example, you specify one hour for your job to run and that job gets routed to a J90, it will receive 4.32 hours of CPU time. When you request CPU time in C90 hours, PBS automatically converts the time if the job ends up on a J90. The *qstat* command displays time in C90 hours. Resource requirements and other specifically designed options determine whether the job can run on a J90. So, precisely specifying job resource requirements will help move your job through the system as fast as possible.

New qsub Resource Attributes

The *qsub* command is used to submit a job to the metacenter queues. Use the "-l" flag and one or more of the system resource attributes to identify your job requirements (see the man pages or online web documents for more information). Several new resource attributes that simplify metacenter usage have been added to the list of those recognized by *qsub*. Table 1 explains these attributes, which are also described online in the man page "pbs_resources."

Attribute	Description
host=hostname	Specifies the target host. Valid hosts are eagle, newton1, newton2, newton3, and newton4.
arch={c90,j90}	Refers to architecture; currently only "c90" and "j90" are valid.
software=s	Qualifies a job based on software required to run to completion. Valid values for "s" are limited to software that is not available on all platforms. The qstat -S command displays a list of known software dependencies. Additions to the list will be made on request; however, these will be limited to general-use software, *not* personal code.
mincpus=n ncpus=n	Specifies the minimum number of CPUs required to run the job. Used in conjunction with ncpus, allows users to provide a range of possible destinations for a job based on the number of CPUs available. The following describes the behaviors of the possible combinations of these two attributes: • Both set; the job will run on the first available system with at least mincpus and no more than ncpus. The system environment variable, NCPUs, is set to the minimum of physical CPUs or ncpus. • Only ncpus set; the job will run on the first available system with at least ncpus CPUs available. NCPUs is set to ncpus. This is the exact number case. • Only mincpus set; the job will run on the first available system with at least mincpus available. NCPUS is set to the physical number of CPUs on the chosen system. • Neither set; the job will run on the first available system. NCPUS is set to the minimum physical number of CPUs on the chosen system and the default for the queue. *qstat -fQ queuename* will display these values.

*Table 1.*New resource attributes for the qsub command.

New qstat Flags

The qstat command has several new flags that provide a variety of useful information. These are described in Table 2.

An excerpt from the *qstat -Aq* command, shown in Table 3, gives detailed information provided queue by queue, system by system. By comparing this information to a user's job resource requirements, the PBS software can route a job to any metacenter system.

Flag	Description
-Aq or -AQ	Shows the resource limits for each queue on each system.
-S	Displays the list of software available for selection on the software dependency list.

*Table 2.*New qstat command flags.

Queue	Memory (Mw)	CPU Time	Walltime	Node	Run	Que	LM	State
batch_eagle	64	08:00:00	--	--	0	0	--	E R
batch_newton 4	64	08:00:00	--	--	0	0	--	E R
batch_newton 2	125	08:00:00	--	--	0	0	--	E R
batch_newton 3	64	08:00:00	--	--	0	0	--	E R
batch_newton 1	125	08:00:00	--	--	0	0	--	E R
mtask_eagle	64	16:00:00	08:00:00	--	0	0	2	E R
mtask_newton 4	64	16:00:00	08:00:00	--	0	0	1	E R
mtask_newton 2	250	16:00:00	08:00:00	--	0	0	2	E R
mtask_newton 3	64	16:00:00	08:00:00	--	0	0	1	E R
mtask_newton 1	220	16:00:00	08:00:00	--	0	0	2	E R

*Table 3.*Excerpt from the qstat -Aq command.

A Few Other Changes

The following is a brief list of additional changes that are part of the metacenter implementation:

· The highbatch queue (now called hbatch) has been incorporated into the metacenter. Each user will get two one-hour "buckets" to run short jobs with quicker turnaround. Jobs are limited to 30 minutes each.

· One set of queues exists for all metacenter systems.

· Home file systems are cross mounted; for example, users can see Eagle's home file system on Newton1.

· The realtime queue name has changed to rtime.

Special Considerations

To enable a job to run on either architecture, a copy of the program must be available that has been compiled for that architecture. In addition, system-specific resources such as $FASTDIR cannot be used. $BIGDIR, on the other hand, is available on both architectures and is especially beneficial when running on the Newtons. The home file systems from both Eagle and the Newtons will be available wherever the job runs. Running from an NFS-mounted file system has particularly bad effects on overall system performancetherefore, $BIGDIR should be used.

One scenario would be to have identical directory trees set up for job execution, so that the program executables are in the same place. Then, no matter where the job runs, the suitable program executable is used.

An alternative method would be to determine the architecture using the *uname* command, which returns a string that can be tested for the string "C90" or "J90." For example:

eagle% uname -m

Cray C90

A simple test from within a script might be:

if ('uname -m | awk '{ print $2 }'' == "C90") then

runC90

else

runJ90

endif

Regardless of the method used to determine which architecture the job is running on, data files should be copied to $BIGDIR to ensure they reside on a local file system. Experience with NFS-mounted file systems clearly shows this to be a benefit. (For more information, see "Users Win by Utilizing $BIGDIR on the Cray," from *NAS News*, September-October '97.)

Not all users will directly "see" the benefit of the Cray metacenter. For example, if a job requires more memory than Eagle has available, the job must run on the Newtons. Or, if job performance requires $FASTDIR, the job can't run on the Newtons. However, by making thorough use of all of the metacenter resources, overall turnaround and throughput are likely to improve. For those of you who can run jobs on either architecture, the positive effects should be noticeable.

NASnews
Main Menu
May–June 1998 • Volume 3, Number 3

Setting Up Secure Shell on Your Unix Accounts

by Ayse Sercan

If you read the article "Secure Shell: New Measure to Keep Accounts Safe" in the January-February issue of *NAS News*, you already know that remote users can now use Secure Shell (SSH) to encrypt their connections. This client-server program, developed at Helsinki University of Technology in Finland, uses a highly secure public-private key encryption scheme to prevent access to servers by unauthorized clients. If you haven't read the article, you should -- because within the next year the NAS Facility will require all remote users to use SSH to reduce the risk of security breaks.

Whichever group you belong to, however, you might appreciate a few practical hints about using SSH. This article explains how to set up SSH on a Unix shell account (see "Getting to the Root of the Matter," below) and how to connect to NAS systems using SSH from a Unix workstation connected to the Internet.

Download, Compile, Install

First, you need to figure out whether SSH is installed on your system. Your system administrator should know, but you can find out by trying the *slogin* command to connect to a remote system. If SSH is already installed but you don't know where, type "whereis slogin" to find its path. Internal NAS system users have it easy -- their workstations are already running the latest version of SSH (1.2.22), and they just have to add "/usr/prg/bin" to their paths.

If your system doesn't have SSH, you or your system administrator can get it easily over the Internet. The Secure Shell home page points to FTP servers for downloading source code or precompiled binaries. Note that legitimate copies of SSH are authenticated with Pretty Good Privacy (PGP), as explained on the Secure Shell website. Be sure to download a legitimate copy of SSH before using it. (For information on using PGP, see Network Associates' web page.

The downloaded file will have been archived with *tar* and compressed with *gzip*. To decompress and unarchive it, type:

```
gzip -c -d ssh-1.2.22.tar.gz | tar xvf -
```

(For more information on using these commands, type "man gzip" or "man tar" to consult the online manual.)

When you've finished unarchiving the file, go to the ssh-1.2.22 directory and follow the instructions in the INSTALL file, which lists what you need in order to complete the installation and gives basic step-by-step instructions and troubleshooting help. The file also explains how to customize your setup so that SSH is automatically invoked, even if you use *rlogin* instead of the SSH-specific command *slogin*. For further troubleshooting and installation information, read the SSH FAQ.

Creating Public and Private Keys

When SSH is installed, it automatically creates a key identifying the system it's on. You can leave it at that, and simply use SSH as a replacement for the Berkeley "r" commands. Or, save yourself some effort by letting the SSH key exchange process take care of verifying your identity, rather than having to enter a password every time you log on to a remote system. To do this, you will need to generate a personal key.

At the prompt, type "ssh-keygen." The program will ask where it should place the key (the default works fine), and then asks for a passphrase. A passphrase is a string of easy-to-remember words like "secure shell is your friend." The software will then create one file called ~/.ssh/identity.pub, which contains your public key for accessing systems with SSH, and another called ~/ssh/identity, which contains your private key (*handle this file with care*). In order to access a system that's running SSH, add the text in your identity.pub file to the ~/.ssh/authorized_keys file. (If that file doesn't already exist, create it using the contents of your identity.pub file.)

To add your key to a system at the NAS Facility, such as Eagle or Chuck, just create a ~/.ssh/authorized_keys file in your home directory on that system. The key must be on a single line, so if you use a text editor to create this file, make sure it doesn't word-wrap or add hard returns.

Once you've created public and private keys, SSH will not only encrypt your username and password (which are sent as plain text by rlogin and telnet), but it can spare you the trouble of typing your password every time you connect to NAS systems. It accomplishes this through ssh-agent, a program designed to hold SSH authentication keys. Typically, ssh-agent is started at the beginning of an X-window or login session and sets two shell environment variables (SSH_AUTHENTICATION_SOCKET and SSH_AGENT_PID) that are inherited by programs started later.

To add private keys to ssh-agent, use ssh-add; programs started after you've added a key can use that key. Then, when you want to log into a system where you are already an authenticated user (that is, your public key is stored in that system's ~/.ssh/authorized_keys file), the software will automatically and invisibly authenticate your identity, so you no longer need to enter a password. (Beware, however: if you're using ssh-agent and fail to secure your workstation, anybody can walk up and connect to any system on which you are an authenticated user.)

Automatic authentication also makes it possible to use SSH to execute single commands on remote systems without even opening a full session. The syntax for doing so is:

```
ssh remote-host-name command
```

where "remote-host-name" is replaced with
the name of the remote system (for example, *chuck.nas.nasa.gov*), and "command" is replaced with the command you wish to run (for example, *who*, or *ls ~/bjones/code-samples/*).
The output from the command will appear,
and then the connection to the remote host will be dropped.

Getting to the Root of the Matter

The above guidelines assume that you have root privileges on your Unix system, or that SSH was installed by somebody who does. But you can still use SSH even if you don't have access to the root directory or your system administrator can't install the software for you. In that case, simply run an SSH binary in your home directory and use it to connect to remote systems such as those at the NAS Facility.

You can also run sshd (the SSH server software) from your home directory, supplying the option
-p port-number (for example, "sshd -p 1024") so it binds to a nonprivileged port (port 1024 or higher), and then connect from another system to that account by typing the command *ssh -p port-number*. This trick will only allow connections into your account on that system, and if the system has to be rebooted, you'll have to restart SSH manually. If the remote system doesn't allow detached processes, you're out of luck.

More Information On the Web

You can find more information on installing and customizing SSH through the Secure Shell home page, or on Kimmo Suominen's informative web page "Getting Started With SSH". For more information on using SSH to connect to NAS systems, contact the NAS security team.

NASnews
▶ Main Menu
May–June 1998 • Volume 3, Number 3

Scientific Visualization Grants Awarded

by Arsi Vaziri

The NAS data analysis group recently issued a NASA Research Announcement (NRA) titled "Research in Scientific Visualization of Computational Fluid Dynamics and Related Aerosciences." The intent of the NRA is two-fold: to foster mutually beneficial technology transfer for scientific visualization, and to solicit basic and applied research proposals that augment research being done in the group.

The NRA sought proposals from a wide range of interest areas, including studies of new visualization techniques such as feature detection, multi-source and comparative visualization, interactive and collaborative interfaces, and integration of numerical simulation and data analysis.

Staff members in the data analysis group evaluated the proposals based on relevance to the group's objectives and goals, overall scientific and technical merit, and degree of unique and innovative methods, approaches, and concepts. The review committee also considered the experience and qualifications of the investigators, the required research facilities, and cost. After thorough review and evaluation, the group selected four proposals for funding. The winners are as follows:

Towards Real-time Vector Field Visualization for Massive and Multi-Source Data Using Hierarchies

Bernd Hamann, Principal Investigator, Center for Image Processing and Integrated Computing, University of California, Davis. This project is aimed at the development of next-generation technology for the visualization of massive data originating from a variety of sources. The data might be empirical or might be the result of a simulation, with or without an associated mesh topology. The ultimate goal of the research is the development of a prototype multi-source data visualization system allowing real-time navigation by using a precomputed data hierarchy, through tessellation of space.

Accelerated Time-Varying Flow Visualization with Feature Preservation

Allen Van Gelder and Jane Wilhelms, Principal Investigators, University of California, Santa Cruz. This project will develop tools for data reduction, analysis, and visualization of large, complex, multidimensional datasets, with emphasis on vector flow fields in time-varying computational fluid dynamics (CFD). In particular, the project will explore methods based on stream surfaces for computation, reorganization, and visualization of flow data. Investigators will also develop decimation and multi-resolution techniques for a feature-based comparison of datasets and images.

Comparative Visualization of Experimental Wind Tunnel Data and Computational Fluid Dynamics Simulations

Alex Pang, Principal Investigator, University of California, Santa Cruz. This project focuses on visualization techniques for comparing and integrating multi-source data from wind tunnel measurements and CFD simulations. Because some of the wind tunnel measurements are image based, this project will investigate image-based, data-level, and feature-based comparative visualization methods.

Physical Presence and Multiple Viewers for Virtual Model Displays

Ian McDowall, Principal Investigator, Fakespace Inc., Mountain View, CA. This is a research and development project for passive and active props to be developed as interaction tools for use with virtual model displays built on the virtual workbench technology. The props are intended to create intuitive interfaces for the direct manipulation and control of elements of 3D visualization environments. Physical items created using the props may also provide enhanced collaborative interfaces. The implementation and evaluation of this research will be integrated with the Virtual Wind Tunnel software at NAS.

Progress Continues on Current Grants

In addition to the new grants and the contract that will be awarded based on this NRA, other scientific visualization research projects supported by the data analysis group are in progress. Active grants scheduled to be completed during 1998 include:

· "Multiresolution and Explicit Methods for Vector Field Analysis and Visualization," Gregory Nielson, Principal Investigator, Arizona State University.

· "Tools for Analysis and Visualization of Large Time-Varying CFD Datasets," Jane Wilhelms and Allen Van Gelder, Principal Investigators, University of California, Santa Cruz.

· "Automated Fluid Feature Extraction from Transient Simulations," Robert Haimes, Principal Investigator, Massachusetts Institute of Technology.

Several agreements are also in place for collaborative research with university faculty, including :

· Principal investigators Arie Kaufman of the State University of New York (SUNY) at Stonybrook for "Time-Critical Volume Rendering";

· Hector Garcia-Molina of Stanford University for "Representation and

Integration of Scientific Information";

· Alex Pang at the University of California, Santa Cruz for "Visual Methods for Model and Grid Validation";

· and Pei Cao at the University of Wisconsin, Madison, for "Study of Memory System Performance for Fluid Flow Post-Processing Tools."

Lists of current and past recipients of NAS-sponsored visualization grants are available on the World Wide Web or by contacting Arsi Vaziri.

In this issue...

Recent NAS-hosted Technical Seminars Available on Video

Within This Article...

Several technical seminars are presented each month at the NAS Facility. Seminars held during the first quarter of this year are described below. Videotapes of most seminars are available on loan. Send a request to the NAS Documentation Center at doc-center@nas.nasa.gov. Inquiries about NAS training events should be directed to Marcia Redmond at redmond@nas.nasa.gov.

A Comparison of Automatic Parallelization Tools/Compilers on the SGI Origin2000 Using the NAS Benchmarks. Writing parallel code by hand is time-consuming and costly. At the March 24 New Technology Seminar, Michelle Hribar, NAS Systems Division, suggested a method for automating the process by porting codes using parallelization tools and compilers. She compared the performance of the handwritten NAS Parallel Benchmarks with: CAPTools, an interactive computer-aided parallelization tool that generates message passing code; the Portland Group's HPF compiler; and using compiler directives with the native FORTRAN77 compiler on the Silicon Graphics Inc. Origin2000.

Large-scale Parallel Computations. At the March 12 New Technology Seminar, Suhrit Dey, from the Mathematics Department at Eastern Illinois University, discussed how large-scale nonlinear models can be subdivided into a number of nonlinear subsystems. These subsystems can then be solved in parallel by applying a Newton-type algorithm.

Entropy and Information in Neural Spike Trains. The nervous system represents time-dependent signals in sequences of discrete, identical action potentials or spikes. Information is carried only in the spike arrival times. But how does one quantify this information in bits, free from any assumptions about which features of the spike train or input signal are most important? At the February 23 New Technology Seminar, Steven Strong, Institute for Advanced Study at Princeton University, explained an approach that can be applied to the analysis of experiments on a variety of systems. He explained that, in order to identify the important statistical features of neural spike trains, the information in these systems must be quantified.

Molecular Electronics: An Ab Initio Approach. Jorge Seminario, Department of Chemistry and Biochemistry at the University of South Carolina, discussed an ab initio approach to molecular electronics at the February 19 NAS Nanotechnology Seminar. Molecular-scale electronics is a field of study that proposes the use of single molecules as the key components in future computational devices. Seminario noted that single molecules which have strategically placed electronic transport barriers could serve as switches and logic devices. He said that although many obstacles remain, an increase in computing performance opens up the possibility of
molecular-scale electronic architectures for future ultracomputing.

Quantum-classical Studies of the Interactions of Small Molecules with Metal Surfaces. In another Nanotechnology Seminar on February 19, Perla Balbuena, Department of Chemical Engineering at the University of South Carolina, discussed some of the studies done on the interactions of small molecules with metal surfaces. Balbuena discussed ab initio techniques used to study the interactions of water and its products of dissociation with nickel surfaces. Using a self-consistent polarizable continuum model, Balbuena investigated the solvent effects on the optimized parameters. A recently developed technique that characterizes ionic diffusion in aqueous solutions was used to investigate transport properties. Potential applications were also discussed.

Effective Visualization of Large, Multidimensional Datasets. On February 10, Christopher Healey, Computer Science Department at the University of California, Berkeley, presented approaches for effectively visualizing large, multidimensional datasets. He presented results from a number of related areas, such as computer graphics, databases, and cognitive psychology, and discussed two aspects of the multidimensionality problem. Healey gave an overview of each area and presented the results from each, addressing the problem of multidimensional data visualization. He discussed how issues in graphics and cognitive psychology can impact research in scientific visualization and gave examples of how theoretical results have been applied to real-world visualization problems.

Image-based Modeling and Rendering at U.C. Berkeley. In another talk held on February 10, Paul Debevec, also from the Computer Science Department at the University of California, Berkeley, discussed new rendering techniques that use view-dependent projective texture mapping to produce renderings in real time. Debevec presented two applications of these techniques.

Analysis and Synthesis of Subwavelength Diffractive Optical Elements for Integration with Active Devices. Dennis Prather, Electrical Engineering Department at the University of Delaware, discussed the numerical electromagnetic modeling of diffractive optical elements (DOEs) at the January 29 New Technology Seminar. Prather explained that, although these elements must be integrated with active components to enable the widespread use of optoelectronics in interconnects and sensor applications, designing a DOE is difficult because of the micron scale of the active components and the micron scale of their operating wavelength. Prather discussed methods used in the design and analysis of micron-scale DOEs, as well as the results of high efficiency lenses and beam splitters made especially for integration with active devices.

Algorithmic Multithreading Programming in Cilk. Charles Leiserson, from MIT's Laboratory for Computer Science, talked about algorithmic multi-threaded programming in Cilk at the January 20 New Technology Seminar. The Cilk language, currently being developed at MIT, minimally extends the C language to allow programmers to specify the interactions among computational threads in a high-level fashion. Leiserson said that MIT's implementation of Cilk is highly efficient.

The Character of Nanoscale Features in Reactor Pressure Vessel Steels Under Neutron Irradiation. At the January 7 Nanotechnology Seminar, Brian Wirth, Department of Mechanical and Environmental Engineering at the University of California, Santa Barbara, talked about the character of nanoscale features in reactor pressure vessel steels under neutron irradiation. He discussed the need for a detailed atomic-scale understanding of the fine-scale features that cause irradiation embrittlement of nuclear reactor pressure vessel steels. Wirth suggested atomic-scale computer simulation to predict this embrittlement and described these computer simulation results.

In this issue...

Credits

Executive Editor: Thomas Lasinski

Editor: Jill Dunbar

Staff Writer: Wade Roush

Contributing Writers: William Chan, George Myers, Ayse Sercan, Arsi Vaziri

Image Coordinator: Chris Gong

Online Page Layout and Graphics: Joel Antipuesto, Chris Gong, Eunah Choi, Rosemary Wadden, Peter Adams

Other Contributors: Tyler Allison, Richard Anderson, Robert Beidron, Nick Cardo, Justin Collins, James Donald, Tom Faulkner, Gail Felchle, Steve Heistand, Mary Hultquist, Randy Kaemmerer, David Kenwright, Scott Lawrence, Lynda Leeser, Alan Powers, Marcia Redmond, Jim Stobie, Jim Taft, Jim Weilmuenster, Chris Williams, Alex Woo, Sarita Wood, Susan Ying

Editorial Board:
Cristy Brickell, Jill Dunbar, Chris Gong, Thomas Lasinski, Nateri Madavan, Patrick Moran, George Myers, Wade Roush, Harry Waddell